Рисунок 6 – Информация о залах

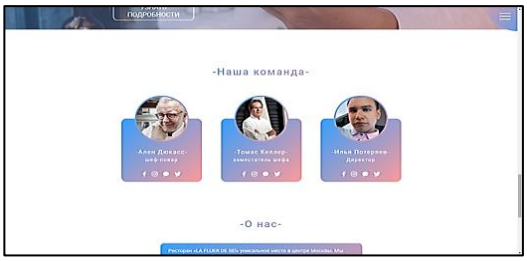

Рисунок 7 – Информация о команде

Моменты разработки. Описанный блок с услугами ресторана реализован через функционал PopUp [2]. Другими словами, это выплывающие окно, расположенное на самом блоке. Простейшие анимации, расположенные в середине страницы – это те же блоки, но в уменьшенном размере, в параметрах которых описано движение [3].

Важную часть составляет сам сервер с базами данных на нем, в которых хранится информация о пользователях. Для реализации сервера использовалась портативная среда OpenServer и приложение phpMyAdmin. Как видно из названия, для работоспособности и взаимосвязи сервера с сайтом используется скриптовый язык PHP [4]. Его задачей будет связь с базой данных на сервере, получение информации. На языке PHP первый скрипт написан для подключения и получения доступа в базу данных. Данный язык очень хорошо предусмотрен для работы с базами данных. В самом скрипте мы обращаемся к БД [1], к таблицам и ее полям по имени, сохраняя туда введённую пользователем информацию с сайта. Это реализовано в онлайн-бронировании столика. Второй же скрипт предназначен для получения информации с базы данных и отображения ее на странице сайта. Как можно понять, это сделано для отображения комментариев. То есть вся информация, которую пользователь предоставляет сайту, хранится безопасно в базе данных.

Заключение. Таким образом, взяв за основу способ «версткой блоков» вместе с дизайном UX/UI и проанализировав результат, можно выстроить теорию о наилучшем построении структуры сайта. Также напрашивается вывод, что каждый сайт, на котором пользователь предоставляет свои данные, должен хранить их на безопасном сервере в базах данных. В современном мире ни один хороший сайт не обходится без сервера.

Литература

1 Зандстра, М. Методики PhP + SQL / М. Зандстра. – Москва : Эксмо, 2013. – 415 с.

2 Дакетт, Дж. Основы веб-программирования с использованием HTML, XHTML и CSS / Дж. Дакетт. – Москва : Эксмо, 2010. – 768 с.

3 Глинников, М. Уроки Web – мастерства. Урок 3. Эскиз первой страницы. Работаем с HTML вручную / М. Глинников // Мир ПК. – 2003. – № 4. – 648 с.

4 Алексеев, А. П. Введение в Web-дизайн : учеб. пособие / А. П. Алексеев. – Москва : Солон-Пресс, 2008. – 356 с.

УДК 519.2

А. В. Потехин

ИССЛЕДОВАНИЕ СЕТИ МАССОВОГО ОБСЛУЖИВАНИЯ С ОГРАНИЧЕННЫМ ВРЕМЕНЕМ ПРЕБЫВАНИЯ ЗАЯВОК В УЗЛАХ

Статья посвящена исследованию стационарного функционирования двухузловой марковской сети массового обслуживания с ограниченным временем пребывания заявок

в узлах. Составлены уравнения глобального и локального равновесия. Показано, что рассматриваемая марковская цепь является регулярной, консервативной и неприводимой. Установлено условие эргодичности, найдено единственное стационарное распределение вероятностей состояний, совпадающее с эргодическим.

Математическая теория массового обслуживания является разделом теории случайных процессов, которая изучает класс задач, возникающих на практике. Моделируются ситуации, связанные с обработкой поступающих требований, заявок к обслуживающему устройству. В настоящее время модели теории массового обслуживания находят широкое применение в компьютерных сетях, сетях передачи данных и т. д. [1].

Сетью массового обслуживания называется совокупность одновременно функционирующих систем массового обслуживания, в которой заявки перемещаются в соответствии с матрицей маршрутов.

Исследована открытая марковская сеть с двумя узлами, в которую поступает пуассоновский (простейший) поток заявок с параметром λ. Все заявки входного потока направляются в первый узел. Времена обслуживания заявок в различных узлах не зависят от процесса поступления заявок и имеют показательное распределение с параметрами $\mu_i, i = 1,2$. Время пребывания заявки в i-м узле ограничено случайной величиной условное распределение которой (если в i-м узле находится n_i заявок) показательное с параметром $\frac{v_i}{n_i}$, $i = 1,2$ [2]. После окончания обслуживания заявки в первом узле она переходит на обслуживание во второй узел. После окончания обслуживания заявки во втором узле она покидает сеть. После окончания времени пребывания заявки в первом узле она покидает сеть. Аналогично, после окончания времени пребывания заявки во втором узле с вероятностью 0,5 заявка покидает сеть, либо с вероятностью 0,5 заявка переходит в очередь первого узла. Функционирование сети описывается марковским процессом n(t).

Схема сети представлена на рисунке 1.

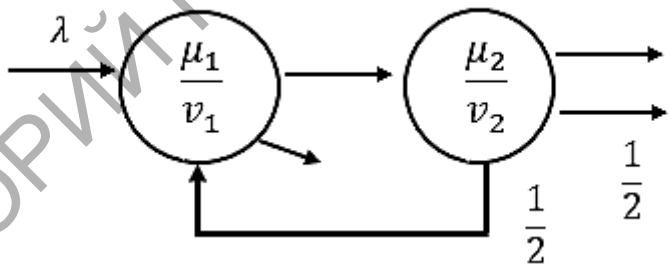

Рисунок 1 – Схема сети для процесса n(t)

Для марковского процесса составлено следующее уравнение глобального равновесия:

$$p(n_1,n_2)\left(\lambda+(\mu_1+v_1)I_{\{n_1\neq 0\}}+(\mu_2+v_2)I_{\{n_2\neq 0\}}\right)=$$
$$=\lambda p(n_1-1,n_2)I_{\{n_1\neq 0\}}+v_1(n_1+1)p(n_1+1)+$$
$$+\left(\mu_2(n_2+1)+\frac{1}{2}v_2(n_2+1)\right)p(n_1,n_2+1)+$$
$$+\mu_1(n_1+1)p(n_1+1,n_2-1)I_{\{n_2\neq 0\}}+$$
$$+\frac{1}{2}v_2(n_2+1)p(n_1-1,n_2+1)I_{\{n_1\neq 0\}}.$$

Для составления уравнений локального равновесия глобальное уравнение разбивается на локальные уравнения путем приравнивания соответствующих слагаемых:

$$p(n_1, n_2)\lambda = v_1 p(n_1 + 1, n_2) + \left(\mu_2(n_2 + 1) + \frac{1}{2}v_2(n_2 + 1)\right)p(n_1, n_2 + 1);$$

$$p(n_1, n_2)(\mu_1 + v_1) = \lambda p(n_1 - 1, n_2) + \frac{1}{2}v_2 p(n_1 - 1, n_2 + 1);$$

$$p(n_1, n_2)(\mu_2 + v_2) = \mu_1 p(n_1 + 1, n_2 - 1).$$

Так как получившиеся уравнения описывают процесс размножения и гибели, то в i-ом узле стационарное распределение имеет вид:

$$p_i(n_i) = \rho_i^{n_i}(1 - \rho_i), n_i = 1,2, \ldots,$$

где $\rho_i = \frac{\lambda\varepsilon_i}{\mu_i + v_i}$, ε_i – решение уравнения трафика, $i = 1,2$.

Уравнения трафика запишутся в виде:

$$\begin{cases} \varepsilon_1 = 1 + \frac{1}{2}\frac{\varepsilon_2 v_2}{\mu_2 + v_2}, \\ \varepsilon_2 = \frac{\varepsilon_1 v_1}{\mu_1 + v_1}. \end{cases}$$

Решая систему относительно ε_1 и ε_2, получим

$$\begin{cases} \varepsilon_1 = \frac{2(\mu_1 + v_1)(\mu_2 + v_2)}{2(\mu_1 + v_1)(\mu_2 + v_2) - \mu_1 v_2}, \\ \varepsilon_2 = \frac{2\mu_1(\mu_2 + v_2)}{2(\mu_1 + v_1)(\mu_2 + v_2) - \mu_1 v_2}. \end{cases}$$

Предполагаемое стационарное распределение вероятностей состояний сети имеет вид:

$$p(n_1, n_2) = p_1(n_1)p_2(n_2).$$

Для доказательства предположения о виде стационарного распределения проверяем уравнения локального равновесия с учётом решения уравнений трафика и предполагаемого вида стационарного распределения. Получаются тождества, следовательно, стационарное распределение найдено верно [3].

Стационарное распределение p (n_1, n_2) существует и единственно, если выполняется условие эргодичности.

Регулярная, консервативная, неприводимая марковская цепь эргодична тогда и только тогда, когда система уравнений равновесия имеет нетривиальное (ненулевое) решение: $\sum_{i \in X}|x_i| < +\infty$ [4].

Достаточным условием регулярности является: $\exists c \geq 0\ \forall n \in X\ q_n \leq c$.

В нашем случае: $q_n = \lambda + \mu_1 I_{\{n_1 \neq 0\}} + \mu_2 I_{\{n_2 \neq 0\}} \leq \lambda + \mu_1 + \mu_2 = c$. То есть условие регулярности выполняется.

Для проверки консервативности составим интенсивности переходов для процесса $n(\mathrm{t})$. Положим, что $\vec{n} = (n_1, n_2)$; $\overrightarrow{e_1} = (1,0)$; $\overrightarrow{e_2} = (0,1)$. Интенсивности переходов процесса $n(t)$ имеют вид:

$$q(\vec{n}) = \lambda + (\mu_1 + v_1)I_{\{n_1 \neq 0\}} + (\mu_2 + v_2)I_{\{n_2 \neq 0\}};$$

$$\begin{cases} q(\vec{n}, \vec{n} + \overrightarrow{e_1}) = \lambda, \\ q(\vec{n}, \vec{n} - \overrightarrow{e_1}) = v_1 I_{\{n_1 \neq 0\}}, \\ q(\vec{n}, \vec{n} - \overrightarrow{e_2}) = (\mu_2 + \frac{1}{2} v_2) I_{\{n_2 \neq 0\}}, \\ q(\vec{n}, \vec{n} - \overrightarrow{e_1} + \overrightarrow{e_2}) = \mu_1 I_{\{n_1 \neq 0\}}, \\ q(\vec{n}, \vec{n} - \overrightarrow{e_2} + \overrightarrow{e_1}) = \frac{1}{2} v_2 I_{\{n_2 \neq 0\}}, \\ q(\vec{n}, \vec{m}) = 0, \text{для остальных.} \end{cases}$$

Сложив все уравнения в системе, получим $q(\vec{n})$, то есть цепь Маркова является консервативной.

Для проверки неприводимости построим цепочки из состояния (0,0) в (n_1, n_2) и обратно:

$$(0,0) \overset{\lambda}{\to} (1,0) \overset{\lambda}{\to} (2,0) \overset{\lambda}{\to} \dots \overset{\lambda}{\to} (n_1 + n_2, 0) \overset{\mu_1}{\to} (n_1 + n_2 - 1,1) \overset{\mu_1}{\to} (n_1 + n_2 - 2,2) \overset{\mu_1}{\to} \dots \overset{\mu_1}{\to} (n_1, n_2).$$

Значит, состояние (n_1, n_2) достижимо из (0,0). Аналогично

$$(n_1, n_2) \overset{\mu_1}{\to} (n_1 - 1, n_2 + 1) \overset{\mu_1}{\to} (n_1 - 2, n_2 + 2) \overset{\mu_1}{\to}$$
$$\dots \overset{\mu_1}{\to} (0, n_1 + n_2) \overset{\mu_2}{\to} (0, n_1 + n_2 - 1) \overset{\mu_2}{\to} (0, n_1 + n_2 - 2) \overset{\mu_2}{\to} \dots \overset{\mu_2}{\to} (0,0),$$

то есть состояние (0,0) достижимо из (n_1, n_2).

Так как все состояния достижимы из нулевого, то есть в любое состояние (n_1, n_2) можно перейти из нулевого (0,0) и, наоборот, в нулевое можно перейти из любого состояния путем поступления, обслуживания и ухода заявок из сети, то отсюда следует неприводимость.

Проверим выполнение условия теоремы Фостера [3].

$$\sum_{n_1=0}^{\infty} \sum_{n_2=0}^{\infty} p_1(n_1) p_2(n_2) < +\infty.$$

Так как n_1 не зависят от n_2 получаем:

$$\sum_{n_1=0}^{\infty} p_1(n_1) \sum_{n_2=0}^{\infty} p_2(n_2) = \sum_{n_1=0}^{\infty} p_1(n_1) \sum_{n_2=0}^{\infty} p_2(n_2).$$

Учитывая стационарное распределение в i-ом узле, получим:

$$\sum_{n_1=0}^{\infty} \rho_1^{n_1} (1 - \rho_1) \sum_{n_2=0}^{\infty} \rho_2^{n_2} (1 - \rho_2).$$

Для того, чтобы сходилось произведение рядов, нужно, чтобы сходился каждый ряд. Эти ряды будут сходиться, если:

$$\rho_i < 1, i = 1,2.$$

В этом случае члены рядов будут являться бесконечно убывающими геометрическими прогрессиями, а они сходятся, если их знаменатель q меньше 1, в нашем случае $q = \rho_i < 1, i = 1,2$, то есть сходимость доказана.

Это условие и есть искомое условие эргодичности. Если это условие будет выполняться, то будет существовать единственное стационарное распределение, совпадающее с эргодическим.

Литература

1 Малинковский, Ю. В. Теория массового обслуживания : учебное пособие по спецкурсу / Ю. В. Малинковский, А. Д. Буриков, М. А. Маталыцкий. – Гродно : Издательский центр ГрГУ, 1984. – 106 с.

2 Гнеденко, Б. В. Сети массового обслуживания с ограниченным временем ожидания в очередях / Б. В. Гнеденко // АВТ. – 1985. – № 2 – С. 50–55.

3 Уолрэнд, Дж. Введение в теорию массового обслуживания / Дж. Уолрэнд. – Москва : Мир, 1993. – 336 с.

4 Kelly, F. P. Networks of Quasi-Reversible Nodes / F. P. Kelly // Adv. Appl. Probab.- Comp.Sci.: Proc. Of the ORSA-TIMS BRS. – Boston, 1981. – P. 147–168.

УДК 004.415.2:004.032.6:004.8:004.43Python

А. М. Протченко

РАЗРАБОТКА ГОЛОСОВОГО АССИСТЕНТА «SEVERITY»

Статья посвящена разработке голосового помощника с использованием языка программирования Python. Разработанное приложение позволяет управлять функциями компьютера посредством голоса. Программа предоставляет возможность работы с файлами в программах пакета MS Office, а также в других текстовых редакторах, позволяет работать с браузером, открывать наиболее распространенные сайты.

Голосовой помощник – программное обеспечение, позволяющее управлять мобильным устройством или компьютером посредством голосовых команд. Голосовые ассистенты способны значительно облегчить некоторые бытовые вопросы и, например, сократить время на поиск и обработку информации. Наиболее известными представителями голосовых помощников являются Алиса от Яндекса, Siri от Apple, Google Assistant для OS Android, Cortana для устройств, поддерживающих Windows.

Для разработки голосового ассистента необходимо соответствующее программное обеспечение. Для создания данного проекта был выбран язык программирования Python, т. к. это высокоуровневый язык программирования, который имеет в своем распоряжении огромное количество библиотек для работы над задачами в различных направлениях. Также использовались технологии распознавания и преобразования речи в текстовый формат для дальнейшей обработки команд. Для реализации пользовательского интерфейса использовался графический модуль PyQt. Для конфигурации приложения использовался файл в формате YAML. Голосовой ввод осуществляется исключительно на английском языке. Это связанно с тем, что в разработке использовался модуль SpeechRecognizer, который на данный момент не адаптирован для других языков.

Принцип работы голосового помощника «Severity» следующий: при запуске выполняется инициализация настроек из стартового конфигурационного файла, далее осуществляется запуск программы для работы, после чего возможен голосовой ввод команд (если команда существует в списке предлагаемых команд, то она выполняется, иначе на экране выдается соответствующее сообщение).

Начальное меню при запуске приложения показано на рисунке 1.