

АЛГОРИТМЫ ПОСТРОЕНИЯ КАРКАСОВ СЕТЕВЫХ АРХИТЕКТУР ЗАДАННОЙ СЛОЖНОСТИ

Одна из задач обучения имитационному моделированию - обеспечить полный цикл работ, включая разработку модели и ее аттестацию. Важнейший этап - мониторинг системы для получения ее параметров и характеристик. Трудности обучения на реальных системах, потребность в формировании описаний параметров большого числа учебных систем, трудоемкость проверки их корректности и получения эталонных характеристик делает актуальной задачу автоматического формирования параметров систем заданной сложности и режима функционирования.

Типовая система состоит из узлов, обеспечивающих обслуживание и перемещение потоков запросов. Они поступают извне, образуя входные потоки. Система организует процессы обслуживания в соответствии со своим законом функционирования. Процессы в общем случае носят вероятностный характер. Система описывается набором параметров структуры и процессов [1], включая состав узлов $B = \{ b, | i = \overline{1, N} \}$; матрицу связности $D = [d_{ij}]$, где $d_{ij} = \{0; 1\}$ в зависимости от наличия связи i -го узла с j -м; канальности узлов K_i , быстродействия каналов V_j , и другие; матрицы P переходов запросов из узла i в узел j ; параметры законов поступления и обслуживания запросов в узлах сети.

Генерация СА сводится к "комбинаторному" получению каркасов сетей [2], снабжаемых далее всеми необходимыми параметрами [1]. Стратегия применения алгоритмов генерации СА базируется на дереве сложности архитектур, описывающем иерархию важности параметров сети по критерию достижения оптимальных результатов генерации.

Порядок получения спецификаций. Первоначально генерируются матрицы D , описывающие структуру сети и удовлетворяющие входным ограничениям. Для фиксированного числа узлов выполняется перебор возможных вариантов их расстановок, обеспечивающих однократный проход из источника в приёмник сети через каждый узел. Каждый вариант расстановки даёт один вариант КС, который хранится в сжатом формате Dc , отображающем матрицу D с помощью одномерных структур. КС представляется последовательностью номеров всех однократно проходимых узлов сети x_i , что отображается размещением $Dc = \{x_i\}_0^{N+1}$. Для получения матричного представления КС, отображающего D , используются данные о связях узлов d_{ij} .

Соответственно в каждом элементе Dc запоминается номер смежного узла, а в структуре данных P_L формируется список последовательно расположенных узлов каркаса. Построение КС реализуется рекурсивным алгоритмом, где на каждом этапе рекурсивного вызова для узла каркаса u ищется номер смежного узла x . При этом проверяется корректность смежности узлов КС, которая нарушается, если узел x предшествует узлу u , если образуется петля или переход из текущей вершины в конец КС.

Каждый каркас проверяется на корректность, которая нарушается, если не выполняется условие соответствия КС, приведенное выше, что на соответствующем шаге генерации приводит к рекурсивному откату назад на предыдущий шаг формирования КС. Или если превышено максимальное количество прямых либо обратных связей

$(N + 1) \cdot N / 2$. Проверка КС производится алгоритмом обхода графа (представленного матрицей D), где k – номер шага, x – номер вершины, Nv – список посещённых вершин.

Каркасы оснащаются заданным числом прямых и обратных связей, что реализуется добавлением к сети вероятностных узлов. Для этого: описание КС переводится в матричный формат D ; в матрице D случайным образом выбирается номер строки i из диапазона $(0, N-1)$; производится поиск узла i в ранее полученном списке вершин P_L . Номера вершин до найденной включительно дают конечные точки обратных связей, а после найденной – конечные точки прямых связей. На пересечении найденного столбца и строки ставится единица.

Полученная матрица D является прообразом P и может содержать в строке от одной и более единиц, что указывает на ветвление маршрутов. На ее основе генерируются значения элементов матрицы P . Так для переходов, помеченных в D нулем, в соответствующий элемент P записывается: нулевая вероятность; остальные значения формируются с учетом нормирующего условия $1 = \sum_{j=0}^N p_{ij}$ (где $i = \overline{1, N}$) случайным образом как

$$p_{ij} = \begin{cases} \text{rand}(0,1), & \text{если } s_{ij} = 1, \\ 0, & \text{если } s_{ij} = 0. \end{cases}$$

Аналитически определяются параметры узлов КС. В том числе: – по среднему значению времени m_i между запросами в потоке, взятому из заданного диапазона, определяется интенсивность поступления запросов $\lambda_0 = 1 / m_i$; – построенная в соответствии со свойством линейности сети [1] система $\lambda_i = \sum_{j=0}^N \lambda_j \cdot p_{ji}$ (где $i = \overline{1, N}$) дает значения интенсивности потоков в узлах сети $\{\lambda_i\}$; – интенсивности используются для определения быстродействий каналов, расчета времени обработки запросов в каналах $t_j = \rho_j \kappa_j / \lambda_j = \rho_j \kappa_j / \lambda_0 \alpha_j$ устройств.

Значения остальных параметров КС генерируются случайным образом из заданных диапазонов или вариантов. Дополнительная проверка СА производится имитационным моделированием полученных описаний и оценкой их характеристик. По результатам генерируются отчёты, предназначенные для организации обучения – содержат описания СА с проверочной информацией, отчеты для обучаемых – с вариантами описаний СА.

Подход макетировался применительно к генерации параметров сетевых архитектур (СА) для систем, описываемых в терминах сетей массового обслуживания, были предусмотрены проектные решения по расширению данного подхода на стохастические сетевые модели, а также по визуализации сетевых архитектур. Система реализована на языке C++ с использованием библиотеки линейной алгебры uBLAS. Пользовательский интерфейс, генерация html-отчётов реализованы на языке C++ с использованием кросс-платформенного инструментария QT. Разработана иерархия классов, поддерживающая хранение результатов и генерацию спецификаций СА. Система поддерживается большинством ОС типа Windows, Linux, MacOS путём перекомпиляции без изменения исходных кодов, требует около 600 Кб памяти.

Таким образом, сформулированы требования к спецификациям сетевых архитектур, согласованные с задачами организации имитационного моделирования. Предложен подход к автоматизации их генерации с учетом требуемой сложности сетей. Рассмотрены алгоритмы получения каркасов сетей и определения их параметров. Приведены ре-

зультаты макетирования. Система обеспечивает генерацию спецификаций сетевых архитектур с сохранением результатов в XML базе данных и генерацию html-отчетов с описаниями СА, с тестовой информацией по данным из XML базы данных.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Ивницкий, В.А. Теория сетей массового обслуживания / В.А. Ивницкий. – М.: Физико-математическая литература, 2004. – 772 с.
2. Андерсон, Д. Дискретная математика и комбинаторика / Д. Андерсон. – СПб.: Вильямс, 2004. – 960 с.

УДК 004.514.62

Никонюк А.Н.

Научный руководитель: к.т.н, доцент Костюк Д.А.

РЕАЛИЗАЦИЯ МОДУЛЯ РАСШИРЕНИЯ ДЛЯ ДИНАМИЧЕСКОГО НЕЛИНЕЙНОГО МАСШТАБИРОВАНИЯ В ОКОННОМ МЕНЕДЖЕРЕ COMPIZ

Задача манипулирования окнами в ситуации, когда их совокупная площадь существенно превышает разрешение устройства вывода, возникла почти одновременно с графическим интерфейсом пользователя. Среди способов облегчить навигацию в таких условиях можно упомянуть прокрутку рабочего пространства, позволяющую видеть его по частям, и иконификацию – «сворачивание» окна в пиктограмму, снабженную поясняющей подписью. С ростом вычислительных возможностей процессоров и разрешающей способности экранов появились графические оболочки и приложения, отображающие вместо пиктограммы скрытого окна его уменьшенное изображение. Хотя такой подход сопряжен с рядом технических трудностей, возможность видеть одновременно, хотя бы с уменьшенной детализацией, изображение всего рабочего пространства дает пользователю ощутимые преимущества [1, 2]. Особенно это актуально в среде динамически изменяющихся объектов, к которым относятся все многозадачные графические среды современных операционных систем.

В последнее время проблема недостатка рабочей области получила дополнительное развитие из-за роста популярности портативных устройств – нетбуков и планшетных компьютеров, способных в той или иной степени запускать приложения, интерфейс и модель взаимодействия с пользователем в которых изначально рассчитаны на стандартное разрешение и размер экрана. Подобные устройства оказываются не только не способны разместить на экране нужное количество окон, но зачастую не могут показать целиком одно стандартное окно.

Нами разработана модель нелинейного масштабирования окон и реализующий ее модуль расширения для оконного менеджера CompiZ, позволяющие сократить размер занимаемой площади окна произвольного приложения с сохранением читаемости и доступности его содержимого.

Концепция нелинейного масштабирования окон и ее реализация для аппаратно-ускоренного манипулирования окнами

Концепция переменного масштаба смоделирована нами в виде деления рабочей области окна на две зоны: центральную 1 и периферийную 2. В зоне 1 отображается основная информация окна, представленная в единичном масштабе. В зоне 2 выводится изображение, искаженное в соответствии с функцией $a(y)$, монотонно возрастающей