

УДК 681.3

Г. Л. Муравьев, А. Н. Никонюк, В. И. Хвещук

КОМПЬЮТЕРНАЯ ГЕНЕРАЦИЯ СПЕЦИФИКАЦИЙ СЕТЕВЫХ АРХИТЕКТУР ЗАДАННОЙ СЛОЖНОСТИ

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 результаты макетирования

Введение

Одна из задач обучения имитационному моделированию – обеспечить полный цикл работ, включая разработку модели и ее аттестацию [1–3]. 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.

Постановка задачи. Требования к спецификациям

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 $\langle X, H, C \rangle$, где X – параметры окружения, $H = \langle S, F \rangle$ – параметры системы, включая параметры структуры S и процессов F , C – характеристики функционирования [1, 3].

Параметры структуры: состав узлов $B = \{ b_i | i = \overline{1, N} \}$; матрица связности $D = [d_{ij}]$, где $d_{ij} = \{0; 1\}$ в зависимости от наличия связи i -го узла с j -м; параметры узлов $Z = \{ z_i | i = \overline{1, N} \}$, задающие канальность узла K_i , быстродействие канала V_j и другие. Параметры Q процессов: матрицы переходов $\{ P^{(q)} = [p^{(q)}_{ij}] | q = \overline{1, Q} \}$, где значение $p^{(q)}_{ij} = [0, 1]$ – вероятность движения запроса из узла i в узел j ; параметры $\{ \bar{h}_i^{(q)} | q = \overline{1, Q}; i = \overline{1, N} \}$, задающие законы поступления $\{ \bar{f}_t^{(q)} | q = \overline{1, Q} \}$ и законы обслуживания заявок $\{ \bar{f}_{\theta, i}^{(q)} | q = \overline{1, Q}; i = \overline{1, N} \}$ в узлах сети.

Сложность сетей NC может быть ограничена заданием требований к таким параметрам сети, как Q, N , число прямых и обратных связей, типы распределений и другие. Требуемый режим функционирования сети может быть задан значениями коэффициентов загрузки узлов $\{ \rho_i \}$ сети.

Тогда задача сводится к получению значений X, H сетевых архитектур (СА) при соблюдении ограничений на NC . При этом на этапе порождения каркасов структур сетей (КС) определяются параметры B, D, P , отвечающие NC ; на этапе аналитического расчета определяются параметры узлов, обеспечивающие заданный режим функционирования; на этапе вероятностного доопределения уточняются недостающие параметры.

Таким образом, получаемые спецификации сетевых архитектур должны отличаться неповторяемостью и прогнозируемой сложностью, что может быть обеспечено наличием эмпирически, либо математически обоснованных процедур их порождения, правилами хранения и учета.

Соответственно необходимо обеспечить:

- управляемость сложностью СА и их характеристиками путем настройки алгоритмов на заданные ограничения; генерацию внутренних описаний уникальных СА;
- документируемость, генерацию соответствующих отчетов;
- проверяемость результатов.

Общее описание подхода

Генерация СА сводится к «комбинаторному» получению каркасов сетей [4], снабжаемых далее всеми необходимыми параметрами [1, 3].

Стратегия применения алгоритмов генерации СА базируется на дереве сложности архитектур, описывающем иерархию важности параметров сети по критерию достижения оптимальных результатов генерации. Для получения наибольшего числа существенно отличающихся по архитектуре сетей следует двигаться от максимально значимых уровней сложности и переходить на нижние уровни лишь при отсутствии желаемого числа “уникальных” архитектур текущего уровня.

В основу генерации положен смешанный комбинаторно-аналитический подход с рандомизированным доопределением параметров. Он обеспечивает:

- получение КС заданной сложности в одномерном, «сжатом» формате на базе рекурсивного алгоритма перебора и отсеивание некорректных КС;
- получение матричного представления КС и его «оснащение» прямыми и обратными связями;
- вероятностное доопределение параметров матрицы переходов запросов в сети;
- аналитический расчет интенсивностей потоков заявок, параметров узлов, декомпозицию параметров по потокам запросов, вероятностное определение недостающих параметров.

При отсутствии ограничений на число вариантов подход обеспечивает порождение спецификаций всех возможных сетевых архитектур с уникальными каркасами.

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 приемник сети через каждый узел. Каждый вариант расстановки дает один вариант КС, который хранится в сжатом формате D_c , отображающем матрицу D с помощью одномерных структур. КС представляется последовательностью номеров всех однократно проходимых узлов сети x_i , что отображается размещением $D_c = \{x_i\}_0^{N+1}$. Для получения матричного представления КС, отображающего D , используются данные о связях узлов d_{ij} .

Таким образом, полученные КС должны удовлетворять условию

$$\begin{cases} DA = \{x_i\}_0^{N+1}, \forall i, j (i = 1, N+1; j = 1, N+1) | x_i \neq x_j; \\ x_0 = 0, x_{N+1} = N+1; \\ P_{x_i, x_{i+1}} = \text{число из диапазона } (0; 1], \end{cases}$$

где x_i, x_{i+1} – номера смежных узлов D_c .

Соответственно в каждом элементе D_c запоминается номер смежного узла, а в структуре данных P_L формируется список последовательно расположенных узлов каркаса. Построение КС реализуется рекурсивным алгоритмом, приведенным ниже, где на каждом этапе рекурсивного вызова для узла каркаса u ищется номер смежного узла x

ГЕНЕРИРОВАТЬ-КС (D_c, y)

ЕСЛИ не получено необходимое число КС ТО

ЦИКЛ-ПОКА не просмотрены все узлы сети ($x = \overline{1, N+1}$)

ЕСЛИ (СМЕЖНОСТЬ-УЗЛОВ-ПОДТВЕРЖДЕНА (x, y, D_c)) ТО

Корректировать $D_c [y] := x$

ЕСЛИ текущий КС построен ($y = N$) ТО

ЕСЛИ (КАРКАС-ВАЛИДЕН ($D_c, 0, 0, P_L$)) ТО

Запомнить описание D_c, P_L

КОНЕЦ-ЕСЛИ

КОНЕЦ-ЕСЛИ

ГЕНЕРИРОВАТЬ-КС ($D_c, y+1$)

КОНЕЦ-ЕСЛИ

КОНЕЦ-ЦИКЛА

КОНЕЦ-ЕСЛИ .

При этом проверяется корректность смежности узлов КС, которая нарушается, если узел x предшествует узлу y , если образуется петля или переход из текущей вершины в конец КС

bool СМЕЖНОСТЬ-УЗЛОВ-ПОДТВЕРЖДЕНА (x, y, D_c)

ЕСЛИ (($x < y$) И ($D_c [x] = y$)) ИЛИ (($y = 0$) И ($x = N+1$))) ТО

Вернуть результат - ложь

ИНАЧЕ

Вернуть результат - истина

КОНЕЦ-ЕСЛИ .

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, $(N + 1) * N / 2$.

Проверка КС производится алгоритмом обхода графа (представленного матрицей D), где k – номер шага, x – номер вершины, Nv – список посещенных вершин

```
bool КАРКАС-ВАЛИДЕН ( Dc, x, k, Nv )
    Добавить узел x в список Nv
    ЕСЛИ пройдено необходимое количество шагов ( k = N + 1 ) ТО
        ЕСЛИ узлы пройдены по разу
            ( Nv содержит N + 1 уникальный узел ) ТО
                Вернуть результат - истина
            ИНАЧЕ
                Вернуть результат - ложь
        КОНЕЦ-ЕСЛИ
    КОНЕЦ-ЕСЛИ
КАРКАС-ВАЛИДЕН ( Dc, Dc [x], k+1, Nv ) .
```

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 $i = \text{ЧИСЛО-ИЗ-ДИАПАЗОНА} (0, N-1)$; 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 ве-

роятность; остальные значения формируются с учетом нормирующего условия $1 = \sum_{j=0}^N p_{ij}$ (где $i = \overline{1, N}$) случайным образом, как

$$p_{ji} = \begin{cases} \text{rand}(0, 1], & \text{если } s_{ji} = 1; \\ 0, & \text{если } s_{ji} = 0 \end{cases}$$

Аналитически определяются параметры узлов КС. В том числе:

– по среднему значению времени m_τ между запросами в потоке, взятому из заданного диапазона, определяется интенсивность поступления запросов $\lambda_0 = 1 / m_\tau$;

– в соответствии со свойством линейности сети [1] строится система уравнений $\lambda_i = \sum_{j=0}^N \lambda_j \cdot p_{ji}$ (где $i = \overline{1, N}$), решение которой дает значения интенсивностей поступления запросов в узлы сети $\{\lambda_i\}$;

– указанные интенсивности далее используются для определения быстродействий каналов, для расчета значений времени обработки запросов в каналах $t_j = \rho_j \kappa_j / \lambda_j = \rho_j \kappa_j / \lambda_0 \alpha_j$ устройств.

Производится распределение потоков по узлам сети. При этом, если количество вариантов задано строго, то выполняется случайное распределение потоков. В противном случае перебираются все возможные случаи размещения потоков по устройствам.

Значения остальных параметров КС генерируются случайным образом из заданных диапазонов или вариантов.

Если ограничение на число вариантов сетевых архитектур не задано, то подход обеспечивает получение всех возможных СА с уникальными каркасами, т. е. каждой СА соответствует свой КС. Их общее число определяется количеством сочетаний n уникальных каркасов по Q потокам и составляет $C_n^Q = n! / (Q!(n-Q)!)$.

Полученные СА на этапе генерации проверяются на корректность и неповторяемость. Дополнительная проверка производится имитационным моделированием полученных описаний и оценкой их характеристик.

По результатам работы системы генерируются отчеты. Отчеты первого типа предназначены для организации обучения, контроля результатов и содержат текстовые описания СА с проверочной информацией, включающей характеристики сети. Отчеты второго типа предназначены для обучаемых и содержат только варианты описаний СА.

Реализация

Подход макетировался применительно к генерации параметров сетевых архитектур (СА) для систем, описываемых в терминах сетей массового обслуживания [1–3], и может быть распространен на произвольные стохастические сетевые модели.

Система реализована на языке C++ с использованием библиотеки линейной алгебры uBLAS [6] из собрания библиотек Boost. Пользовательский интерфейс, экспорт-загрузка XML, генерация html-отчетов реализованы на языке C++ с использованием кросс-платформенного инструментария QT [5].

Хранение результатов, описаний сетевых архитектур производится в структурированном виде в html-формате в XML-базе данных, что в совокупности с каскадными таблицами стилей CSS обеспечивает генерацию и форматирование отчетов любой сложности и совместимость при передаче данных другим системам обработки информации.

Разработана иерархия классов, поддерживающая хранение результатов и генерацию спецификаций СА, включая базовый класс, агрегирующий классы генератора КС, генератора параметров КС, класс настроек генератора.

Система поддерживается большинством ОС типа Windows, Linux, MacOS путем перекомпиляции без изменения исходных кодов, требует около 600 Кб памяти. При динамической сборке для работы системы должна быть установлена библиотека QT, либо должны присутствовать необходимые библиотечные файлы. Например, для Windows это файлы QtCore4.dll, QtGui4.dll, libgcc_s_dw2-1.dll, mingw10.dll.

Заключение

Таким образом, система обеспечивает: генерацию спецификаций сетевых архитектур сетей массового обслуживания по заданным ограничениям с сохранением результатов в XML базе данных; генерацию полных человекочитаемых html-отчетов с описаниями СА, с проверочной, тестовой информацией по данным из XML базы данных.

Хранение результатов в XML базе данных обеспечивает совместимость по данным с другим системами, а использование HTML, CSS позволяет генерировать отчеты любой сложности.

Литература

1. *Ивницкий, В. А.* Теория сетей массового обслуживания / В. А. Ивницкий. М.: Физико-математическая литература, 2004.
2. *Советов, Б. Я.* Моделирование систем / Б. Я. Советов, С. А. Яковлев. М.: Высшая школа, 2001.
3. *Рыжиков, Ю. И.* Имитационное моделирование. Теория и технологии / Ю. И. Рыжиков. СПб.: КОРОНА, 2004.
4. *Андерсон, Д.* Дискретная математика и комбинаторика / Д. Андерсон. СПб.: Вильямс, 2004.
5. *Бланшет, Ж.* Программирование GUI на C++ / Ж. Бланшет, М. Саммерфилд. М.: КУДИЦ-ПРЕСС, 2007.
6. Boost uBLAS (Basic Linear Algebra Library) online documentation [Electronic resource] / Boost libraries 1.46.1 2011. – Mode of access: http://www.boost.org/doc/libs/1_45_0/libs/numeric/ublas/doc/index.htm. – Date of access: 23.03.2011.

Муравьев Геннадий Леонидович, профессор кафедры интеллектуальных информационных технологий Брестского государственного технического университета, кандидат технических наук, доцент, mgl_work@mail.ru

Никонюк Александр Николаевич, студент четвертого курса факультета электронно-информационных систем Брестского государственного технического университета, nikoniuk@mail.ru

Хвещук Владимир Иванович, профессор кафедры интеллектуальных информационных технологий Брестского государственного технического университета, кандидат технических наук, доцент, hv_i@tut.by