



ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ФУНКЦИОНИРОВАНИЯ ВЫЧИСЛИТЕЛЬНЫХ ПОДСИСТЕМ ТРЕНАЖЕРНЫХ СИСТЕМ С ПОМОЩЬЮ СЕТЕЙ ПЕТРИ-МАРКОВА

А. Н. ПРИВАЛОВ
Д. В. ЖУКОВ

*Тульский артиллерийский
инженерный институт*

e-mail:
alexandr_prv@rambler.ru

e-mail: den755@rambler.ru

Для моделирования информационных процессов в вычислительных подсистемах тренажерных систем обосновано применение сетей Петри-Маркова (СПМ). Ввиду высокой трудоёмкости разработки и аналитического решения задач предлагается использовать СПМ для получения численного решения путем имитационного моделирования вычислительных систем. Приведены система исходных данных и алгоритмы имитационного моделирования с применением аппарата СПМ.

Ключевые слова: тренажёрная система, система с распределенной обработкой данных, информационные процессы, показатель эффективности тренажёрных систем.

В условиях возрастания возможностей и сложности различного рода технических систем становятся актуальными вопросы подготовки специалистов-операторов по управлению такими системами. Традиционно, тренажеры и тренажёрные системы (ТС) служат основным инструментом подготовки и переподготовки персонала, особенно в военной области.

В настоящее время решение проблемы проектирования ТС сопряжено с разработкой средств информационной поддержки функционирования динамической обучающей среды. Повышение качества информации, предъявляемой оператору в таких системах, является актуальным и требует учёта ряда факторов, таких как: распределённая обработка данных, жёсткие временные ограничения получения ответа на запрос при принятии решения, большие объёмы данных для визуализации обучающей среды и др.

Вычислительные сети с распределенной обработкой данных (СРОД) находят всё большее применение в различных областях человеческой деятельности, в том числе и при подготовке высококвалифицированных специалистов систем «человек-машина», работа которых связана со сложными техническими, опасными для жизни системами. В связи с этим большое внимание уделяется разработке вычислительных подсистем тренажерных систем (ВПТС) на базе СРОД.

Процесс функционирования ВПТС на базе СРОД отличаются высокой степенью сложности, а характеризующие их показатели, такие, например, как средние значения времени получения ответа на заявки абонентов различных категорий, зависят от весьма большого числа факторов. Вместе с тем значительные материальные затраты на создание и эксплуатацию подобных систем побуждают к получению количественных оценок раз личных аспектов функционирования рассматриваемых систем вычислительных средств с целью выбора рациональных инженерных решений на различных этапах их проектирования, эксплуатации, модернизации и развития.

В этой связи особенно большое значение приобретает проблема разработки комплекса математических моделей, адекватно описывающих информационные процессы, протекающие в вычислительных средствах.

Анализ реализации информационных процессов в ВПТС на основе вычислительной сети с распределённой обработкой данных позволяет сделать вывод о том, что для информационно-вычислительных работ характерен высокий уровень параллелизма, то есть одновременного решения различных частей одной вычислительной задачи несколькими процессорами одного или нескольких компьютеров.



С одной стороны, это обусловлено самим сетевым характером вычислительной среды тренажёрных систем, а с другой необходимостью решения ряда сложных в вычислительном смысле задач в реальном масштабе времени.

Указанные соображения выдвигают необходимость обоснования и применения особого подхода к моделированию информационных процессов, протекающих в такой системе.

В качестве такого подхода предлагается использовать сети Петри-Маркова.

Для распределённых вычислительных систем интуитивно можно предположить, что время реализации информационного процесса может варьироваться от величины, получающейся в случае, если все операторы алгоритма последовательно интерпретируются одним процессором, а остальные компоненты системы в это время выполняют другую работу (или простаивают), до величины, получающейся, если все компоненты начинают и заканчивают интерпретацию своих частей алгоритма одновременно и при решении задач исключены случаи их простоя. Указанное обстоятельство и возможность разделения алгоритма на одновременно выполняемые операции порождает предпосылки для оптимизации временной сложности алгоритмов, реализуемых в вычислительной сети с распределённой обработкой данных. В [1] на основании исследования процесса выполнения команды процессором фон-Неймановской ЭВМ показано, что количество машинных тактов, затрачиваемое процессором на её выполнение, является случайной величиной, распределение которой зависит как от особенностей аппаратных средств, так и от распределения обрабатываемых командой данных. Кроме того, в [2] был исследован характер перехода между операторами алгоритма и показана его квазистохастичность. Из квазистохастичности переходов и случайности времени выполнения операторов исследуемых алгоритмов можно заключить, что естественной моделью для описания интерпретации последовательности команд процессорами фон-Неймановского типа является полумарковский процесс [3], определяемый пятёркой

$$M = \{A, Z, q, p, f(t)\},$$

где $A = \{a_{1(a)}, \dots, a_{j(a)}\}$ – конечное непустое множество состояний, совпадающее с множеством операторов алгоритма;

$Z = \{z_{1(z)}, \dots, z_{j(z)}, \dots, z_{j(z)}\}$ – конечное непустое множество переходов между состояниями, совпадающее с множеством переходов алгоритма;

$p = [p_{j(a)j(a)}] - J(a) \times J(a)$ – мерная вложенная цепь Маркова;

$f(t) = [f_{j(a)j(a)}(t)] - J(a) \times J(a)$ – мерная матрица плотностей распределения времени пребывания процесса в состояниях;

$q = [q_{j(a)}] - J(a)$ – мерный вектор начального распределения вероятностей. Начальными (поглощающими) состояниями полумарковского процесса являются состояния, совпадающие с подмножеством начальных (конечных) операторов алгоритма.

При моделировании систем с распределённой обработкой данных необходимо учитывать следующие их особенности:

- определённая и специфицированная для каждой системы СРОД стратегия использования ресурсов для обработки информации;
- динамический характер высвобождения (задействования) вычислительных ресурсов в процессе решения конкретных задач;
- наличие эффекта «соревнования» между параллельно функционирующими компонентами.

Указанные особенности учитываются при моделировании систем с помощью сетей Петри.

Алгоритмы информационных процессов моделируются сетями Петри естественным образом: множество позиций сетей совпадает с множеством переходов между операторами алгоритма, а операторы моделируются переходами.



Причинно-следственный характер связей между позициями и переходами сетей Петри является предпосылкой для моделирования во-первых – структур алгоритмов, а во-вторых- логики событий, происходящих в СРОД. Однако существенным недостатком сетей рассматриваемого вида является их асинхронность. Это обстоятельство не позволяет без принципиальных доработок использовать их для исследования поведения СРОД во временной области. В настоящее время известен ряд попыток приспособить сети Петри и им подобные модели для анализа производительности ЭВМ и сетей, но они не могут быть названы успешными, во-первых, потому что обладают малой степенью общности, а во-вторых, исходные данные для анализа в указанных моделях слабо коррелированы с реальными характеристиками элементарных процессов в компонентах систем и назначаются на основании опыта эксплуатации аналогичных комплексов, экспертных оценок и т.п.

Объединение двух подходов к моделированию информационных процессов в вычислительной подсистеме порождает сеть Петри-Маркова (СПМ).

Известно [4], что сеть Петри-Маркова называется структурно-параметрическая модель, заданная парой

$$\theta = \{\psi, \gamma\}, \quad (1)$$

где ψ – множество резидентных свойств (структурно-параметрические характеристики); γ – множество вариационных свойств (характеристики состояния).

Резидентные свойства СПМ, в свою очередь, задаются парой

$$\Psi = \{P, M\}, \quad (2)$$

где P – Сеть Петри; M – случайный процесс.

Сеть Петри P определяет структуру СПМ, а случайный процесс M накладывается на структуру P и определяет временные и вероятностные характеристики СПМ. Вариационные свойства модели раскрываются через четверку

$$\gamma = \{C, \Phi, \Theta, Q\}, \quad (3)$$

где C – вектор раскраски позиций; Φ – вектор разметки; Θ – вектор занятости; Q – упорядоченное множество очередности заявок.

Структура СПМ характеризуется одним из множеств:

$$P = \{A, Z, I_A(Z), O_A(Z)\}, \quad (4)$$

или $P = \{A, Z, I_Z(A), O_Z(A)\}$,

где $A = \{a_{1(a)}, \dots, a_{j(a)}, \dots, a_{l(a)}\}$ – конечное множество позиций;

$Z = \{z_{1(z)}, \dots, z_{j(z)}, \dots, z_{l(z)}\}$ – конечное множество переходов;

$I_Z(A) = \{I_Z(a_{1(a)}), \dots, I_Z(a_{j(a)}), \dots, I_Z(a_{l(a)})\}$, $O_Z(A) = \{O_Z(a_{1(a)}), \dots, O_Z(a_{j(a)}), \dots, O_Z(a_{l(a)})\}$ – соответственно входная и выходная функции позиций;

$I_A(Z) = \{I_A(z_{1(z)}), \dots, I_A(z_{j(z)}), \dots, I_A(z_{l(z)})\}$ и $O_A(Z) = \{O_A(z_{1(z)}), \dots, O_A(z_{j(z)}), \dots, O_A(z_{l(z)})\}$ – соответственно входная и выходная функции переходов.

Введём ряд определений. Так, примитивным будем называть переход $z_{j(z)}$, для входной и выходной функций которого выполняются условия:

$$\mu[I_A(z_{j(z)})] = \mu[O_A(z_{j(z)})] = 1,$$

где $\mu(\dots)$ – мощность соответствующего множества.

Все остальные типы переходов являются непримитивными (НП). Непримитивные переходы образуют подмножество $Z_{zn} \subset Z$. Для них вводится специальная индексация $i(zn)$. Среди НП может быть выделена группа конечных, или поглощающих переходов.

Конечным, или поглощающим, будем называть непримитивный переход $z_{j(zn)} \in Z_E = \{z_{1(E)}, \dots, z_{j(E)}, \dots, z_{l(E)}\}$, для выходной функции которого выполняется условие

$$O_A(z_{j(zn)}) = \emptyset,$$

где $\emptyset = \{\}$ – пустое множество.

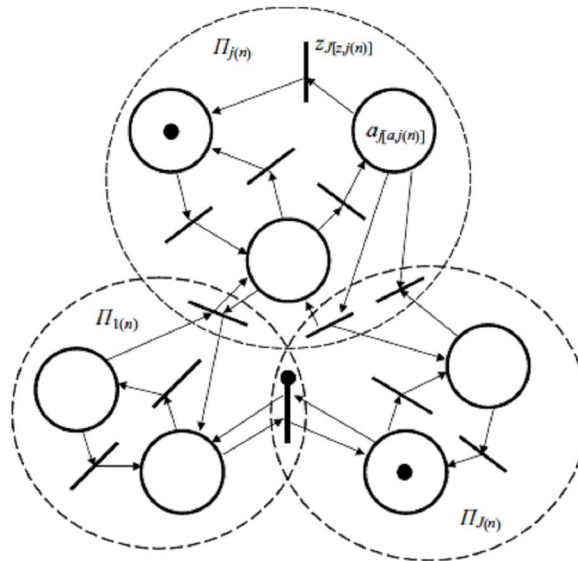


Рис. 1. Сеть Петри-Маркова, разделенная на элементарные подсети

Подсетью Петри-Маркова, называется такая СПМ со структурой $\Pi_{i(m)} = \{A_{i(m)}, Z_{i(m)}, I_A(Z_{i(m)}), O_A(Z_{i(m)})\}$ или $\Pi_{i(m)} = \{A_{i(m)}, Z_{i(m)}, I_Z(A_{i(m)}), O_Z(A_{i(m)})\}$, для которой

$$A_{i(m)} \cap A = A_{i(m)}, \quad Z_{i(m)} \cap Z = Z_{i(m)}, \\ I_A(Z_{i(m)}) \cap I_A(Z) = I_A(Z_{i(m)}), \quad O_A(Z_{i(m)}) \cap O_A(Z) = O_A(Z_{i(m)})$$

или

$$I_Z(A_{i(m)}) \cap I_Z(A) = I_Z(A_{i(m)}), \quad O_Z(A_{i(m)}) \cap O_Z(A) = O_Z(A_{i(m)}).$$

Подсеть Петри-Маркова со структурой $\Pi_{i(m)}$, для которой справедливо, что $I_Z(A_{i(m)}) \cap Z_{i(m)} \neq \emptyset$, $O_Z(A_{i(m)}) \cap Z_{i(m)} \neq \emptyset$, и для любых пар позиций из $A_{i(m)}$ и переходов из $Z_{i(m)}$ которой можно подобрать такие комбинации входных и выходных функций, при которых

$$F_A(\dots F_Z(a_{j[a_j(i)]}) \dots) = a_{j[a_j(i)]} \text{ или } F_Z(\dots F_A(z_{i[z_i(i)]}) \dots) = z_{i[z_i(i)]},$$

где $F = \{1, 0\}$, называется связной подсетью.

Связная подсеть со структурой $\Pi_{i(m)}$, в которой

$$Z_{i(m)} = \{z_{i[m_1, i(m)]}, \dots, z_{i[m_2, i(m)]}, z_{i[z, i(m)]}, \dots, z_{i[m_j, i(m)]}\}$$

и для всех $z_{i[z, i(m)]}$ которой выполняются условия

$$? [I_A(z_{i[z, i(m)]})] = ? [O_A(z_{i[z, i(m)]})] = 1, \quad I_A(z_{i[z, i(m)]}) \cap A_{i(m)} = a_{i[z, i(m)]}, \quad O_A(z_{i[z, i(m)]}) \cap A_{i(m)} = a_{i[z, i(m)]},$$

а для каждого $z_{i[m_n, i(m)]}$ выполняется одно или оба условия

$$I_A(z_{i[m_n, i(m)]}) \cap A_{i(m)} \neq \emptyset, \quad I_A(z_{i[m_n, i(m)]}) \cap A_{i(m)} \neq \emptyset, \quad O_A(z_{i[m_n, i(m)]}) \cap A_{i(m)} \neq \emptyset,$$

называется элементарной подсетью Петри-Маркова (ЭППМ).

В общем случае структура Π связной СПМ может быть представлена в виде объединения ЭППМ: $\Pi = \Pi_{i(m)} \cup \dots \cup \Pi_{j(m)}$ ЭППМ (рис. 1). Таким образом, функциональным подобием ЭППМ является фрагмент алгоритма, интерпретируемый одним из компонентов параллельной вычислительной системы.

Пространством Ω состояний СПМ называется дискретное гиперпространство, ортогональными измерениями которого являются ЭППМ и НП (рис. 2).

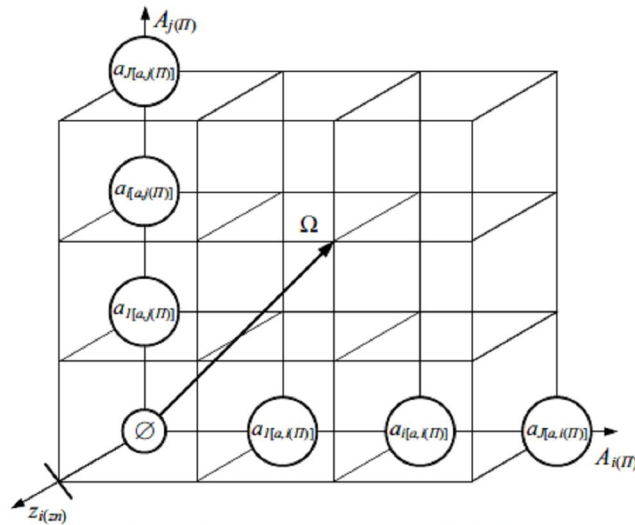


Рис. 2. Пространство состояний СПМ

Дискретными координатами каждого из $J(\Pi)$ измерений пространства Ω , соответствующего ЭППМ, являются позиции ЭППМ. Каждое из $J(zn)$ измерений, соответствующих НП имеет единственную дискретную координату, которой является сам НП. Пересечение измерений дает начало координат, которым является нуль или пустое множество.

В работе [5] показано, что при анализе эффективности функционирования ВПТС на базе СРОД, путем последовательных преобразований СПМ может быть сведена к стартовому, поглощающему переходам и единственной позиции (рис. 3)

$$\Pi = \{a, \{z_b, z_e\}, \{I_A(z_b) = \emptyset, I_A(z_e) = a\}, \{O_A(z_b) = a, O_A(z_e) = \emptyset\}\}. \quad (5)$$

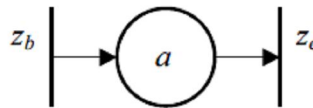


Рис. 3. Итог процедуры последовательных упрощений СПМ

Для позиции a СПМ (5) может быть получено аналитическое выражение для плотности распределения $\phi(t)$ выполнения полушага s_e в поглощающий переход z_e . Далее полученное аналитическое выражение для $\phi(t)$ может быть аппроксимировано аналитическим законом, и для него могут быть найдены математическое ожидание, а также начальный и центральный моменты различных порядков. Указанные выражения и являются оценкой эффективности системы. Например, математическое ожидание плотности распределения может иметь физический смысл времени реализации информационного процесса в вычислительной сети.

Однако подобный подход применим лишь для моделей систем с достаточно простой структурой, поскольку высока трудоемкость получения как самой структурно-параметрической модели, так и ее аналитического решения. В этих случаях рекомендуется применять аппарат СПМ для получения численного решения путем имитационного моделирования вычислительных систем на базе СРОД.

Исходными данными для имитационного моделирования являются:

математическое описание структуры СПМ Π в виде (4);

вектор q вероятностей начала процесса в переходах множества Z в виде



$$q = (q_{1(z)}, \dots, q_{j(z)}, \dots, q_{J(z)});$$

матрица p вероятностей в виде

$$p = (p_{j(a)j(z)});$$

матрица $f(t)$ плотностей распределения времени реализации этапов информационных процессов в виде

$$f(t) = (f_{j(a)j(z)}(t));$$

матрица Λ логических условий выполнения полушагов из переходов множества Z в виде

$$\Lambda = (\lambda_{i(z)i(a)});$$

где $\lambda_{i(z)i(a)}$ – элементы матрицы логических условий, равные

$$\lambda_{j(z)j(a)} = \begin{cases} \lambda[I_A(z_{i(z)})], \text{ если } a_{j(a)} \in O_A(z_{j(z)}); \\ 0, \text{ если } a_{j(a)} \notin O_A(z_{j(z)}). \end{cases}$$

Формируется дискретное пространство с вектором состояний W (рис. 2.)

Сам процесс имитационного моделирования включает в себя ряд алгоритмов, перечисленных ниже.

Алгоритм 1. Определение НП, в котором начинается процесс.

1. Компоненты $q_{1(z)}, \dots, q_{j(z)}, \dots, q_{J(z)}$ вектора q упорядочиваются вдоль числовой оси в интервале $[0, 1]$, и формируются интервалы

$$[0, q_{1(z)}], [q_{1(z)}, (q_{1(z)} + q_{2(z)})], \dots, \left[\sum_{i(z)=1(z)}^{j(z)} q_{i(z)}, \sum_{i(z)=1(z)}^{j(z)+1} q_{i(z)} \right), \dots, \left[\sum_{i(z)=1(z)}^{J(z)-1} q_{i(z)}, \sum_{i(z)=1(z)}^{J(z)} q_{i(z)} \right].$$

2. Запускается генератор случайных чисел и формируется квазислучайное число π , имеющее равновероятный закон распределения, т.е.

$$\varphi(\pi) = \begin{cases} 0, \text{ если } \pi < 0; \\ 1, \text{ если } 0 \leq \pi \leq 1; \\ 0, \text{ если } \pi > 1. \end{cases} \quad (6)$$

3. Выбирается переход $z_{j(z)}$ вектора 0-разметки, имеющего в данном случае вид $\Xi^0 = (\emptyset, \dots, \emptyset, \dots, \emptyset, \mathbf{0}, \dots, \mathbf{1}, \dots, \mathbf{0})$,

в соответствии с правилом: $j(z) = k(z)$, если

$$\sum_{i(z)=1(z)}^{k(z)-1} q_{i(z)} \leq \pi < \sum_{i(z)=1(z)}^{k(z)} q_{i(z)}, \mathbf{1}(z) \leq k(z) \leq J(z).$$

4. Конец работы алгоритма.

Алгоритм 2. Определение направления выполнения полушагов $S_{i(a),i(z)}$ из позиции $a_{i(a)}$ в переходы $O_Z(a_{i(a)})$

Для работы алгоритма 2 предварительно должны быть компоненты $p_{j(a),1(z)}, \dots, p_{j(a),j(z)}, \dots, p_{j(a),j(z)}$ вектора $p_{j(a)}$, представляющего собой $j(a)$ -ю строку матрицы вероятностей p , упорядочиваются вдоль числовой оси в интервале $[0,1]$, и формируются интервалы

$$[0, p_{j(a),1(z)}], [p_{j(a),1(z)}, (p_{j(a),1(z)} + p_{j(a),2(z)})], \dots, \left[\sum_{i(z)=1(z)}^{j(z)} p_{j(a),i(z)}, \sum_{i(z)=1(z)}^{j(z)+1} p_{j(a),i(z)} \right), \dots,$$

$$\left[\sum_{i(z)=1(z)}^{J(z)-1} p_{j(a),i(z)}, \sum_{i(z)=1(z)}^{J(z)} p_{j(a),i(z)} \right].$$



По функциям распределения $F_{j(a),j(z)}(t)$ плотностей распределения $f_{j(a),j(z)}(t)$, представляющих собой $j(a)$ -ю строку матрицы плотностей распределения $f(t)$, строятся обратные функции

$$t_{j(a),j(z)}(\pi),$$

где π – равномерно распределенное в интервале $[0,1]$ квазислучайное число, $t_{j(a),j(z)}(\pi)$ – неслучайная функция случайного аргумента с областью значений $T_{j(a),j(z)} \min \leq t_{j(a),j(z)} \leq T_{j(a),j(z)} \max$.

Алгоритм 2 сводится к выполнению следующих операций.

1. Запускается генератор случайных чисел и формируется квазислучайное число π , имеющее равномерный закон распределения (6).

2. Выбирается направление выполнения полушага $s_{j(a),j(z)}$ из позиции $a_{j(a)}$ в соответствии с правилом: $j(z) = k(z)$, если

$$\sum_{i(z)=1(z)}^{k(z)-1} p_{j(a),i(z)} \leq \pi < \sum_{i(z)=1(z)}^{k(z)} p_{j(a),i(z)} \quad \mathbf{1}(z) \leq k(z) \leq J(z).$$

3. Запускается генератор случайных чисел, и в соответствии с функцией $t_{j(a),j(z)}(\pi)$ определяется момент выполнения полушага $s_{j(a),j(z)}$.

4. Конец работы алгоритма.

Алгоритм 3. Определение направления выполнения полушагов $z_{j(a)}$ из НП $z_{j(z)}$ в позиции $O_A(z_{j(z)})$.

Для работы алгоритма 3 предварительно должны быть сформированы:

наборы булевых констант, соответствующих элементарным конъюнкциям дизъюнктивной нормальной формы булева выражения для $\lambda_{j(z),j(a)}$;

наборы текущих булевых переменных, соответствующих элементарным конъюнкциям дизъюнктивной нормальной формы булева выражения для $\lambda_{j(z),j(a)}$.

Для каждой конституенты единицы с номером $j(l)$ булевы константы $l_{j(z),j(a),j(l),j(l),j(z)}$ в соответствии с выражением:

$$l_{j(z),j(a),j(l),j(l),j(z)} = \begin{cases} 0, \text{ если полушаг } s_{j(l),j(z),j(z)} \text{ согласно } j(l) - \text{й конституенте} \\ \text{не должен быть сделан;} \\ 1, \text{ если полушаг } s_{j(l),j(z),j(z)} \text{ согласно } j(l) - \text{й конституенте} \\ \text{должен быть сделан,} \end{cases}$$

где $j[l,j(z)]$ – индекс, обозначающий номер позиции из входной функции перехода $z_{j(z)}$.

Кроме того, для каждого направления выполнения полушага $j(a)$ из перехода $z_{j(z)}$ должно быть задано количество конституентов единицы $u[j(z),j(a)]$ в дизъюнктивной нормальной форме.

Алгоритм 3 сводится к выполнению следующих операций.

1. После выполнения полушага $s_{j(l),j(z),j(z)}$ во всех наборах текущих булевых переменных логические нули $j[l,j(z)]$ -й переменной меняются на логические единицы.

2. Каждый набор текущих булевых переменных сравнивается с набором булевых констант, соответствующих элементарным конъюнкциям дизъюнктивной нормальной формы булевского выражения для $\lambda_{j(z),j(a)}$.

3. В случае совпадения хотя бы одного набора булевых переменных с набором булевых констант выставляется признак открытия перехода и полушага $s_{i(z),i(a)}$.

4. После выполнения полушагов во все компоненты конституент единицы наборов булевых переменных записывается логический нуль.

Алгоритм 4. Пересчет плотностей распределения после выполнения одного из полушагов из позиции в переход.

Исходными данными для работы алгоритма являются функции плотностей распределения $f_{j(a),j(z)}(t)$ времени выполнения полушагов из позиций $a_{j(a)}$ в переходы



$z_{j(z)} \in O_z(a_{j(a)})$ и интервал времени, отсчитанный от момента выполнения предыдущего полушага T .

Если процессы выполнения полушагов s_φ и s_ψ начинаются одновременно, плотности распределения времени выполнения указанных полушагов равны $\varphi(t)$ и $\psi(t)$, соответственно, то плотность распределения времени ожидания полушагом s_φ события завершения выполнения полушага s_ψ определяется зависимостью

$$f_{1 \rightarrow 2}(t) = \frac{\int_0^\infty \delta(\tau - T_2) \varphi(t + \tau) d\tau}{\int_0^\infty 1(t - T_2) d\Psi(t)} = \frac{1(t) \psi(t + T_2)}{\int_0^\infty 1(t - T_2) d\Psi(t)}, \quad (7)$$

где $1(t - T_2)$ – единичная функция Хевисайда.

Алгоритм заключается в переборе всех функций плотностей распределения и пересчете их по зависимости [7], сводящейся к операции сдвига аргумента в числителе и вычислению вероятности, выражение для которой записано в знаменателе зависимости.

Алгоритмы 1-4 позволяют сформировать нижеследующий алгоритм имитационного моделирования.

Алгоритм 5. Имитационное моделирование ИП с помощью СПМ

1. Устанавливается таймер в положение $t = 0$.
2. Запускается алгоритм 1 и с его помощью формируется 0-разметка СПМ.
3. Параметру k присваивается значение $k = 1$.
4. Из перехода $z_{j(z)}$ выполняются полушаги в позиции $a_{j(a)} \in O_A(z_{j(z)})$, таким образом формируется k -разметка.
5. Перебираются позиции $a_{j(a)}$ и для каждой из них определяется направление выполнения полушагов в соответствии с алгоритмом 2. Для каждого выбранного направления определяется интервал времени выполнения полушага $\Delta t_{j(a)}$.
6. Перебираются интервалы времени $\Delta t_{j(a)}$ и из них выбирается $\Delta t_{j(a) \min}$.
7. Изменяется текущее время $t = t + \Delta t_{j(a)}$.
8. Выполняется полушаг $s_{j(a), j(O, j(a))}$ из позиции $a_{j(a)}$ путем изъятия фишки из позиции $a_{j(a)}$ и помещения ее в соответствующий переход $z_{j(z)} \in O_Z(a_{j(a)})$.
9. Если переход $z_{j(z)} \in Z_E$, то конец.
В противном случае переход к оператору 10.
10. Запускается алгоритм 4 и пересчитываются плотности распределения, для всех позиций, формирующих текущий вектор k -разметки.
11. Запускается алгоритм 3 и проверяются переходы по признаку открытия в связи с выполнением полушага $s_{j(a), j(O, j(a))}$. При отсутствии открытых переходов – переход к оператору 13.
10. При наличии открытых переходов, например $z_{k(z)}$, выполняются полушаги в позиции подмножества $O_A(z_{k(z)})$ и таким образом корректируется вектор k -разметки.
11. $k = k + 1$, переход к оператору 5.

Итогом работы алгоритма 5 является простой статистический ряд временных интервалов $t = \{t_1, t_2, \dots, t_N\}$ достижения подмножества поглощающих переходов Z_E из подмножества начальных. Алгоритм 5 должен быть выполнен достаточное количество раз для того, чтобы набрать удовлетворительную статистику по временным интервалам.

Обработка простого статистического ряда может производиться по известным методикам обработки результатов статистических испытаний, предусматривающих выполнение следующих операций.

1. Разделение всего диапазона $0 \leq t \leq t_{\max}$ на равные участки $0 \leq t < \tau$, $\tau \leq t < 2\tau$, $2\tau \leq t < 3\tau$, ..., $(K - 1)\tau \leq t < t_{\max}$.



2. Определение частот ν_k попадания временных интервалов $t = \{t_1, t_2, \dots, t_n, \dots, T_N\}$ в каждый из K участков по зависимости

$$\nu_k = \frac{\mu\{t \mid (k-1)\tau \leq t < k\tau\}}{N},$$

где $\mu\{\dots\}$ – мощность соответствующего множества.

По частотам ν_k строится гистограмма

$$\gamma = \begin{pmatrix} \frac{\tau}{2} & \dots & \frac{\tau}{2} + k\tau & \dots & \frac{\tau}{2} + (K-1)\tau \\ \nu_0 & \dots & \nu_k & \dots & \nu_{K-1} \end{pmatrix}, \quad (8)$$

где $\frac{\tau}{2} + k\tau$ – середина k -го диапазона, $0 \leq k \leq K-1$; ν_k – частота попадания времени в k -й диапазон.

3. Определение среднего времени достижения поглощающего перехода по формуле

$$T = \frac{\sum_{n=1}^N t_n}{N}. \quad (9)$$

В контексте решаемой задачи указанное время будет являться средним временем реализации информационного процесса.

4. Определение дисперсии случайной величины t по зависимости:

$$D = \frac{\sum_{n=1}^N (t_n - T)^2}{N-1} = \frac{\sum_{n=1}^N t_n^2}{N-1} - \frac{NT^2}{N-1}. \quad (10)$$

Следующей задачей, которая может быть решена при применении имитационного моделирования, является задача аппроксимации гистограммы (8) аналитическим законом распределения. Указанную аппроксимацию целесообразно производить в том случае, если результаты имитационного моделирования по анализируемой модели являются промежуточными, и будут использоваться при аналитическом решении задачи проектирования вычислительной подсистемы более высокого иерархического уровня, где рассматриваемая система является элементом. Вследствие того, что в рассматриваемом случае производится хотя и машинный, но эксперимент, сглаживание плотности аналитическим законом распределения, рассмотренное в предыдущем подразделе, неприменимо. В данном случае целесообразно аппроксимировать гистограмму (8) по критерию χ^2 Пирсона.

Для этого необходимо построить гистограмму (8) графически и подобрать аналитический закон $f(t)$, в наибольшей степени повторяющий форму гистограммы, у которого математическое ожидание совпадает с параметром T , рассчитанным по формуле (9), а дисперсия совпадает с величиной D , рассчитанной по зависимости (10).

Обозначим величины плотностей вероятностей аналитического закона $f(t)$ в

точках $\frac{\tau}{2}, \dots, \frac{\tau}{2} + k\tau, \dots, \frac{\tau}{2} + (K-1)\tau$, соответственно

$$f\left(\frac{\tau}{2}\right) = f_0^*, \dots, f\left(\frac{\tau}{2} + k\tau\right) = f_k^*, \dots, f\left(\frac{\tau}{2} + (K-1)\tau\right) = f_{K-1}^*.$$

Тогда критерий χ^2 для гистограммы (8) и аналитического закона $f(t)$ будет иметь вид:



$$\chi^2 = N \sum_{k=0}^{K-1} \frac{(v_k - f_k^*)^2}{f_k^*}.$$

В соответствии с критерием, по кривой распределения определяется вероятность того, что закон распределения $f(t)$ удовлетворяет критерию близости полученной с помощью имитационного моделирования гистограммы. Если вероятность достаточно велика, то принятая гипотеза о виде и параметрах аппроксимирующего закона распределения не противоречит результатам имитационного моделирования.

Таким образом, СПМ являются универсальным математическим аппаратом, позволяющим получать как теоретические, так и квази-экспериментальные данные о статистических параметрах информационных процессов в вычислительной подсистеме тренажёрной системы.

Литература

1. Игнатъев В.М. Анализ производительности ЭВМ: учеб. пособие/ В.М. Игнатъев, Е.В. Ларкин. – Тула: ТГТУ, 1994. – 104 с.
2. Абдуллаев Д.А. Моделирование локальных вычислительных сетей с учётом вероятностно-временных характеристик/ Д.А. Абдуллаев, У.Б. Амирсаидов //Автоматика и телемеханика. – 1994. – №3. – С. 151-160.
3. Сильвестров Д.С. полумарковские процессы с дискретным множеством состояний / Д.С. Сильвестров. – М.: Сов. радио, 1980. – 272 с.
4. Ларкин Е.В. Сети Петри-Маркова для моделирования параллельных процессов/ Е.В. Ларкин //Приборы и приб. Системы: тез. Докл. Всерос. конф. – Тула: ТулГТУ, 1994. – С. 41.
5. Ларкин Е.В. Проектирование информационных систем роботов с использованием сетей Петри-Маркова: уч.пособие/ Е.В. Ларкин, Н.А. Котова. – Тула: Изд-во ТулГУ, 2008. – 156 с.

SIMULATION FUNCTIONING OF COMPUTING SUBSYSTEMS TRAINING SYSTEMS WITH PETRI NETS-MARKOV

A. N. PRIVALOV
D. V. ZHUKOV

*Tula artillery engineering
institute*

*e-mail:
alexandr_prv@rambler.ru*

e-mail: den755@rambler.ru

To simulate the information process in computing subsystems training systems justified the use of Petri nets, Markov (MTA).

Due to the high complexity of design and analytical problem solving are encouraged to use the MTA to obtain numerical solutions by simulation of computer systems.

Given system input data and algorithms of simulation using the apparatus SPM.

Key words: Virtual training systems, system with distributed data processing, measures of efficiency Virtual training system.