

**ИМИТАЦИОННО-АНАЛИТИЧЕСКОЕ МОДЕЛИРОВАНИЕ
ПРИ МОНИТОРИНГЕ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ¹****А.М. Грузликов, Н.В. Колесов, М.В. Толмачева (Санкт-Петербург)****Введение**

Использование для целей диагностирования информационно-управляющих и вычислительных систем их моделей представляет собой достаточно традиционное научное направление, которому в литературе уделяется значительное внимание [1, 2]. Проблема диагностирования многогранна. Ее решение предполагает использование набора средств, среди которых важное место занимают средства мониторинга вычислений. Причем на практике речь идет, как правило, о мониторинге распределенных, т.е. параллельных вычислений. Целью мониторинга при этом является выявление аномальных событий, список которых далее будет ограничен нарушениями последовательности решаемых задач. Используя в данной работе термин «мониторинг», мы отмечаем тот факт, что причинами аномальных событий в вычислительной системе (ВС) могут быть не обязательно отказы аппаратуры, как это принято в диагностировании, но и допущенные при проектировании ошибки в организации вычислений и разработанных программах.

Будем предполагать, что вычисления в ВС подчинены событийной организации [3]. Последнее означает, что данные обрабатываются программными модулями (ПМ) системы не в соответствии с заранее определенным планом, а по их готовности. Причем в случае, когда исполняющий процессор занят, данные помещаются в соответствующую ему очередь. Формализацией событийной организации могут быть различные имитационно-аналитические модели и, в частности, сеть Петри [4], которая была в свое время предложена именно для описания асинхронного взаимодействия вычислительных модулей. Неудивительно, что при решении задачи мониторинга параллельных вычислений, различным вариантам которого посвящено большое число работ, часто используется именно эта модель [5–10]. При этом объект мониторинга представляется как дискретная событийная система, когда описание системы осуществляется в терминах некоторых происходящих в ней событий.

В нашем случае для целей мониторинга используется имитационно-аналитическая модель системы, а анализируемыми событиями являются прием или выдача информации из программных модулей системы. Предлагаемый подход предполагает введение в анализируемую систему избыточности. Причем особенностью данного случая является то, что избыточность вводится в рамках концепции тестового диагностирования и по существу представляет собой имитационно-аналитическую модель системы.

Авторы используют описываемый подход для систем реального времени, однако он может быть применен и за пределами этого класса систем.

1. Постановка задачи. Для пояснения сути предлагаемого подхода рассмотрим граф межмодульных информационных связей некоторой гипотетической системы S (рис. 1а). Система включает три функционально связанных ПМ F_1 , F_2 и F_3 , которые могут быть размещены как на разных процессорах системы, так и на одном. Каждый из ПМ на основе входных данных (u_1 – для F_1 , u_2 – для F_2 , y_1 и y_2 – для F_3) формирует выходные (y_1 , y_2 и y_3 соответственно) в виде массивов информационных слов. Система мониторинга (СМ), осуществляя тестовый мониторинг системы, формирует для нее тестовые воздействия (информационные слова), дополняя ими входные данные для F_1 и F_2 , и анализирует выходную реакцию ПМ F_3 . Если в каждом из ПМ F_1 , F_2 и F_3 реальные информационные

¹ Работа проводилась при поддержке гранта РФФИ 13– 08– 00211 а.

слова обрабатываются штатными алгоритмами, то параллельно с этим тестовые информационные слова обрабатываются по специальным алгоритмам f_1 , f_2 и f_3 , а результаты их обработки выдаются в составе выходных данных.

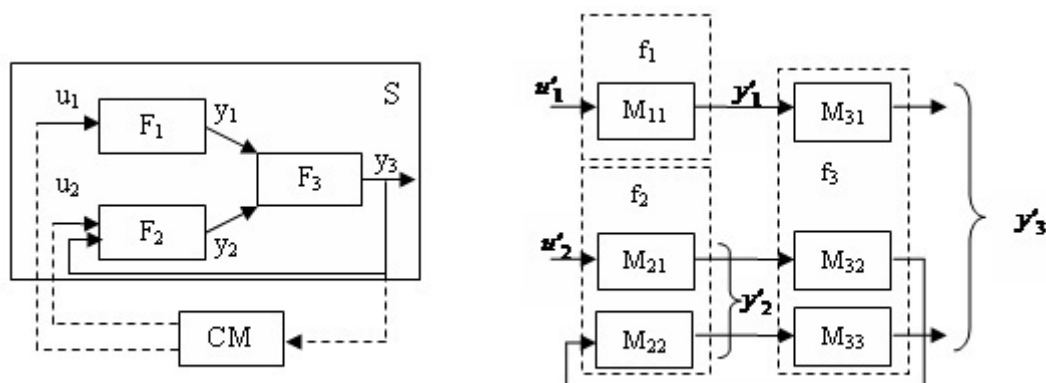


Рис. 1. Система с мониторингом вычислений (а) и ее модель (б)

Задача состоит в том, чтобы путем соответствующего подбора тестовых воздействий и алгоритмов их обработки в ПМ осуществлять мониторинг последовательности обработки тестовых данных, а значит, и мониторинг последовательности обработки реальных данных, поскольку эти два процесса обработки протекают практически синхронно. В определенном смысле алгоритм обработки тестовых данных является имитационно-аналитической моделью алгоритма обработки реальных данных, а значит, и моделью системы. В связи с этим задача синтеза алгоритма обработки тестовых данных может быть сформулирована как задача синтеза модели системы, а в контексте решаемой задачи мониторинга (диагностирования) как задача синтеза диагностической модели. Заметим, что постановка вопроса о синтезе модели достаточно необычна для задач мониторинга (диагностирования). Обычно решается вопрос не синтеза модели, а выбора адекватной модели.

2. Имитационно-аналитическая модель системы. Итак, центральным моментом обсуждаемой задачи является вопрос о надлежащем выборе алгоритма обработки тестовой информации в каждом ПМ. Данный вопрос не является тривиальным. Прежде всего, заметим, что объединенный алгоритм f_{Σ} обработки тестовой информации в модулях системы является композицией частных алгоритмов и реализуется, как и основной алгоритм обработки, в виде параллельного вычислительного процесса в соответствии с событийной моделью. Таким образом, обсуждая далее синтез модели системы, будем иметь в виду синтез событийной модели. С нашей точки зрения, эта модель должна удовлетворять трем условиям. Во-первых, она должна быть такой, чтобы обеспечивать обнаружение рассматриваемых аномальных событий. Во-вторых, она должна быть достаточно простой, чтобы не требовать для своей реализации значительных ресурсов и допускать автоматический синтез для любого графа межмодульных информационных связей. Наконец, в-третьих, анализ ее результатов должен реализоваться в виде последовательного алгоритма или набора таких алгоритмов. По существу, выдвижение всех этих условий связано со стремлением по возможности упростить реализацию мониторинга.

Укажем универсальный подход к автоматическому синтезу f_{Σ} , справедливый для любого графа межмодульных информационных связей. При этом предлагается пойти по следующему пути. Специальными приемами свести рассмотрение информационных графов произвольного вида к анализу некоторых стандартных (примитивных) графов, а

именно, цепей. Найдем по информационному графу множество путей, составляющих покрытие его ребер. Под путем понимается последовательность ребер и вершин, соединяющая некоторый вход информационного графа с его выходом. Для рассматриваемого примера (рис. 1а) покрытие обеспечивается двумя путями с ребрами u'_1, y'_1, y'_3 и $u'_2, y'_2, y'_3, y'_2, y'_3$. Здесь штрихи используются для обозначения передаваемой в массивах тестовой информации. Второй путь содержит цикл для покрытия информационной обратной связи с выхода системы на вход ПМ₂. Сопоставим в модели системы с каждым путем цепь из такого числа динамических звеньев M , через сколько ПМ проходит данный путь. При этом если в путь входит ПМ_{*i*} и он является *j*-м по порядку в этой цепи, то с ним сопоставляется звено M_{ij} . После описанных построений модель системы представляется совокупностью независимых цепей. При этом можно сказать, что результат вычисления модели формируется как массив значений аргументов, подвергнувшихся независимой обработке.

На рис. 1б. представлена модель системы из рассматриваемого примера. При этом модель ПМ₁ содержит одно звено M_{11} , модель ПМ₂ – два звена M_{21} и M_{22} , а модель ПМ₃ – три звена M_{31} , M_{32} и M_{33} . Отметим особенность структуры массива y'_3 , выдаваемого из третьей системы. В ней можно выделить две части, одна из которых формируется в звеньях M_{31} и M_{33} и предназначена для системы мониторинга, другая формируется в M_{32} и предназначена для ПМ₂.

Попробуем в определенной степени уточнить модели звена, ПМ и системы. Из теории технического диагностирования известно [1, 2, 11, 12], что наибольшей эффективности диагностирования при наименьших затратах можно достичь в случае, когда объект диагностирования – линейный. Кроме того, ясно, что, поскольку речь идет об анализе последовательности событий (последовательности решения задач), то этот алгоритм должен быть динамическим. В результате получаем, что моделью звена должна быть линейная динамическая система:

$$x_{i,j}(t+1) = f_{i,j}x_{i,j}(t) + g_{i,j}u_{i,j}(t), \quad y_{i,j}(t) = h_{i,j}x_{i,j}(t) \quad i = \overline{1, m}, j = \overline{1, m_i}, \quad (1)$$

где $x_{i,j}$, $u_{i,j}$ и $y_{i,j}$ – векторы состояния, входа и выхода, $f_{i,j}$, $g_{i,j}$ и $h_{i,j}$ – матрицы динамики входа и выхода *j*-го звена в модели *i*-го ПМ, *m* – число ПМ, *m_i* – число звеньев в ПМ_{*i*}.

Каждое звено модели ПМ обрабатывает информацию лишь от одного массива и по одному и тому же алгоритму. В отличие от звена ПМ в общем случае обрабатывает несколько массивов, а, следовательно, его модель содержит параллельную композицию нескольких звеньев. При этом передаваемые массивы могут иметь разную размерность, определяемую количеством и структурой звеньев, составляющих модель конкретного ПМ. Звенья модели ПМ срабатывают в разные моменты времени и обрабатывают информацию в общем случае по разным алгоритмам. Это означает, что модель ПМ нестационарна:

$$x_i(t+1) = F_i(k)x_i(t) + G_i(k)u_i(t), \quad y_i(t) = H_i(k)x_i(t) \quad i = \overline{1, m}. \quad (2)$$

При конкретизации модели системы учтем, что она состоит из отдельных цепей и, конкретизировав модель цепи, мы в значительной степени конкретизируем модель системы.

Итак, рассмотрим цепь, состоящую из *L* звеньев, каждое из которых описывается уравнениями (1). Представим цепь в виде линейной динамической системы, вектором состояния которой является вектор $x(t)$, составленный из векторов состояния звеньев $x_i(t) \quad i = \overline{1, L}$, входящих в эту цепь.

$$x(t+1) = \tilde{F}(j)x(t) + \tilde{G}(j)u(t), \quad y(t) = \tilde{H}(j)x(t) \quad j = \overline{1, L+1} \quad (3)$$

Эти уравнения описывают L-1 межзвенных обменов и два обмена с СМ (прием и выдача информации). Матрицы в этих уравнениях зависят от номера такта (номера обмена), т.е. модель нестационарна.

Если модель системы содержит M цепей, то ее описание имеет вид:

$$x(t+1) = \tilde{F}_k(j)x(t) + \tilde{G}_k(j)u(t), \quad y(t) = \tilde{H}_k(j)x(t) \quad j = \overline{1, L+1}, \quad k = \overline{1, M}. \quad (4)$$

3. Алгоритм мониторинга системы на основе событийного анализа. Поскольку модель системы распадается на M независимых цепей (4), то, как следствие, и в алгоритме мониторинга можно для каждой из цепей выделить свою составляющую. Ее можно представить как состоящую из генератора тестовой последовательности для цепи и генератора эталонной реакции на эту последовательность. Последний может быть исполнен в виде модели цепи. Таким образом, задача мониторинга системы сводится к мониторингу цепи, что существенно проще, однако и при решении этой задачи также возникают определенные трудности. Действительно, вопросы мониторинга (диагностирования) достаточно подробно изучены для динамических, т.е. последовательных систем. В данном же случае цепь работает как конвейер, т.е. реализует параллельный вычислительный процесс. Сведем ее анализ в целях мониторинга к анализу динамической системы.

Очевидно, что описание цепи в виде динамической системы нельзя считать полностью адекватным, поскольку при этом параллельный процесс описывается как последовательный. Однако решить поставленную задачу мониторинга с использованием такого описания возможно. При этом все звенья цепи заменяются одним звеном с переменными матрицами, повторяющимися с периодом поступления входной информации, т.е. приходим к периодически нестационарному описанию цепи. Для задачи мониторинга важно то, что при номинальном поведении последовательность и содержание выходных сеансов, генерируемых системой и рассчитанных по последовательной модели, будут совпадать, а при рассматриваемых вариантах аномального поведения – расходиться. Безусловно, их временное позиционирование будет различным. Однако этот факт не является препятствием для корректного сопоставления последовательностей в событийной модели, когда эти последовательности представляются очередями, сопоставляемыми посимвольно без привязки к моментам времени их генерации.

Таким образом, каждая цепь модели системы может анализироваться как периодически нестационарная система. При этом в модели системы (4) описание цепи следует рассматривать как описание на периоде длины L+1.

Из теории диагностирования [1, 2] известно, что условием эффективного диагностирования (мониторинга) является наблюдаемость и управляемость диагностируемой системы. Это в полной мере относится и к рассматриваемому случаю, а именно, при синтезе модели системы мы должны обеспечить наличие у нее этих двух свойств. Однако в нашем случае, очевидно, что для наблюдаемости и управляемости модели системы необходимо и достаточно, чтобы каждая из цепей модели была наблюдаема и управляема. Выше было показано, что для описания цепи при решении задачи мониторинга может быть использована модель периодически нестационарной динамической системы. Правила синтеза наблюдаемых и управляемых цепей, а также правила построения тестов для периодически нестационарной динамической системы известны [11, 12].

Заключение. В настоящей работе обсуждаются вопросы мониторинга параллельных вычислений. Предполагается, что вычисления реализуются совокупностью программных модулей и организованы в соответствии с событийной моделью. Процедура мониторинга распадается на совокупность алгоритмов, каждый из которых сопоставляется с вычислительным путем в информационном графе программы. Показывается, что для

целей мониторинга может быть использована модель периодически нестационарной динамической системы.

Литература

1. **Мироновский Л.А.** Функциональное диагностирование динамических систем. М.-СПб.: Изд-во МГУ-ГРИФ, 1998. 256 с.
2. *Patton R. J, Frank P M, Clark R. №.* Issues in fault diagnosis for dynamic systems, Springer-Verlag, London, April 2000.
3. **Топорков, В. В.** Модели распределенных вычислений. – М.: Физматлит, 2004. – 316 с.
4. **Питерсон Д.** Теория сетей Петри и моделирование систем. – М.: Мир, 1984. – 264 с.
5. **Wu Y., Hadjicostis C.** Algebraic approaches for fault identification in discrete-event systems // IEEE Trans. Robotics and Automation, 50(12), 2005, pp.2048–2053.
6. **Cabasino M.P., Giua A., Seatzu C.** Discrete Event Diagnosis Using Petri Nets // ICINCO09: 6th Int. Conf. on Informatics in Control, Automation and Robotics (Milano, Italy), July 2009.
7. **Lefebvre D., Leclercq E., Ould El Medhi S.** Petri net models for detection, isolation and identification of faults in DES // The 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes Barcelona, Spain, June 30 – July 3, 2009, pp. 1372–1377.
8. **Chanthery E., Pencol'e Y.** Monitoring and Active Diagnosis for Discrete-Event Systems // The 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes Barcelona, Spain, June 30 – July 3, 2009, pp. 1545–1550.
9. **Lefebvre D.** Diagnosis with Petri nets according to partial events and states observation // 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFE-PROCESS), August 29–31, 2012. Mexico City, Mexico, pp. 1244–1249.
10. **Cassandras C.G., Lafortune S.** Introduction to discrete event systems. – Dordrecht: Kluwer Academic Publishers. – 2008, p. 848.
11. **Колесов Н.В.** Нестационарная диагностическая модель системы обмена распределенного управляющего комплекса // Автоматика и телемеханика. – 1990. – № 4. – С. 144–154.
12. **Колесов Н.В.** Диагностирование линейных дискретных нестационарных систем // Автоматика и телемеханика. – 1988. – № 7. – С. 157–163.