

УДК 004.052:519.27

О.М. ТАРАСЮК, Л.С. УСОВ, В.С. ХАРЧЕНКО

*Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина*

## МНОГОШАГОВАЯ ПРОЦЕДУРА ПОСТРОЕНИЯ ГРАФА СОСТОЯНИЙ ПРИ ИССЛЕДОВАНИИ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ АППАРАТА МАРКОВСКИХ ПРОЦЕССОВ

Выполнен анализ задач моделирования вычислительных систем (ВС) с использованием аппарата теории марковских процессов. Предложена методика построения графа состояний путем систематизации и пошагового учета допущений, связанных с их функциональностью и надежностью. Приведен пример пошагового построения марковского графа состояний для двухпроцессорной ВС, в которой учитываются различные режимы функционирования и сбои процессоров при выполнении задач

**марковские процессы, граф состояний**

### Введение

**Проблема моделирования вычислительных систем.** При исследовании вычислительных систем (ВС) в процессе их разработки, эксплуатации и реинжиниринга широко используется аппарат теории марковских процессов. Целью такого исследования обычно является анализ зависимости основных характеристик ВС – производительности, надежности, от параметров ее архитектуры, решаемых задач, программных и аппаратных средств, интенсивностей сбоев, отказов.

На основе результатов анализа далее формируются рекомендации по оптимизации архитектуры, режимов функционирования, введения избыточности и т.д. Даже для сравнительно несложных ВС число состояний может исчисляться десятками и сотнями, что характерно для распределенных отказоустойчивых ВС [1]. Это порождает проблему размерности марковской модели и ее адекватности реальной ВС размерности. При этом, на наш взгляд, ключевыми являются три задачи:

1) получение множества и размеченного графа состояний, который в полной мере соответствует ВС, т.е. учитывает особенности функционирования и ограничения, связанные с ее построением и использованием по назначению;

2) подготовка исходных данных (значений параметров) для исследования ВС;

3) выбор и корректное применение стандартных средств компьютерного моделирования на основе программных пакетов Matlab, Mathcad и др. [2].

Первая задача, которая рассматривается далее в статье более детально, включает две подзадачи. Первая из них состоит в формировании и обосновании множества допущений, характерных для исследуемой вычислительной системы, вторая – в определении множества состояний и построении графа состояний в соответствии с принятыми допущениями.

Для сложных систем их решение, как правило, представляет собой итеративный процесс, не гарантирующий быстрого и безошибочного результата.

**Анализ литературы.** В известных работах [3 – 6] используются различные подходы к решению проблемы размерности и безошибочности получения графа состояний, основанные на принципах укрупнения состояний, регуляризации, многофрагментности и др. В частности, принцип многофрагментности позволяет эффективно исследовать ВС с учетом изменяющихся параметров потоков отказов и восстановлений вследствие дефектов программных средств [5, 6].

Графо-событийный анализ с использованием этого принципа удачно применяется при моделировании многоверсионных ВС как средство регуляризации описания и уменьшения размерности графа для их поэтапного проведения исследования [7]. Однако, в известных работах, по нашему мнению, недостаточно внимания уделено первой подзадаче в части систематизации (иерархизации) допущений, позволяющей делать более формализованным и верифицируемым процесс построения графа состояний.

**Цель данной статьи** – совершенствование методики построения марковского графа состояний ВС путем систематизации и пошагового учета допущений различного уровня иерархии.

### Методика построения графа состояний

Суть усовершенствования методики состоит в определении и детализации базовых процедур получения размеченного графа состояний при исследовании сложной ВС, среди которых ключевой является процедура пошагового формирования множества допущений. Методика ориентирована на моделирование отказо- и сбоеустойчивых ВС, в которых нарушение функционирования парируется специальными программными или аппаратными средствами [8].

Граф состояний предлагается строить в следующей последовательности.

1. Определяются группы допущений, касающиеся различных свойств системы и режимов ее функционирования. При этом первоначально целесообразно выделить две основные группы допущений, связанные с ее функциональностью (ДФ) и надежностью (ДН).

2. Формируется список допущений, образующих множество ДФ = {ДФ<sub>1</sub>, ..., ДФ<sub>Г</sub>}:

а) обобщенные допущения – высокоуровневые высказывания об особенностях функционирования системы;

б) детализирующие допущения – допущения, которые конкретизируют обобщенное допущение до такой степени, когда исключается возможность его неоднозначного толкования;

в) ограничивающие допущения – допущения, которые ограничивают область действия обобщенного допущения.

3. С учетом особенностей выполнения системой заданных функций и полученного множества ДФ формируется множество состояний СФ и строится исходный граф ГФ.

4. Производится проверка марковости полученного графа и осуществляется его разметка. В результате получают граф ГФР.

5. Формируется список допущений, образующих множество ДН = {ДН<sub>1</sub>, ..., ДН<sub>К</sub>}.

6. С учетом возможных отказов компонент системы при выполнении различных функций и процедур их восстановления формируется дополнительное множество состояний СН и строится расширенный граф ГФН.

7. Производится проверка марковости полученного графа и выполняется его доразметка. Получают граф ГРФН.

8. Осуществляется анализ полученного графа ГРФ с целью выявления идентичных состояний и их объединения с учетом сохранения марковости.

Следует отметить, что:

– во-первых, множества ДФ и ДН могут быть иерархизированы и вследствие этого, операции по п.п. 3 – 7 выполняют итеративно;

– во-вторых, при формировании допущений целесообразно применять следующую последовательность:

а) формулировка обобщенного допущения;

б) формулировка всех детализирующих допущений;

в) формулировка всех ограничивающих допущений;

г) принятие всех ограничивающих допущений;  
 д) поочередная замена ограничивающих допущений на детализирующие и их отображение на марковской модели;

– в-третьих, состояния из множеств СФ, СН следует обозначать, исходя из множества признаков, которыми они характеризуются. К таким признакам могут относиться: выполнение программ, обмен данными, ожидание; работоспособное состояние, сбой, отказ, выполнение программы восстановления и т.д.

Такой подход позволяет частично формализовать процесс формирования множества состояний и проверки его правильности.

### Пример применения методики

**Описание системы.** Рассмотрим особенности применения методики на примере. Система состоит из двух процессоров и модуля памяти, связанных одной магистралью (рис. 1).

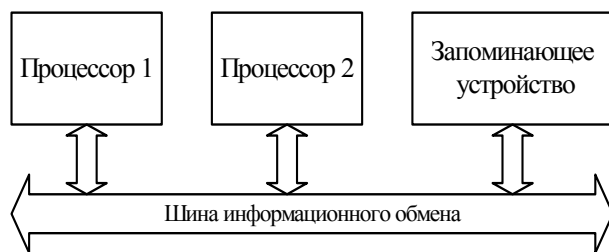


Рис. 1. Пример ВС

Периодически процессоры обращаются к памяти. Если один процессор захватывает магистраль, то второй ждет, пока она не освободится, затем производит обмен с памятью.

В ходе функционирования системы возможны сбои процессоров. Если сбой происходит во время обращения к магистрали, то обращение повторяется после восстановления.

Подробнее проанализируем работу процессоров. У каждого из них есть только два режима работы:

самостоятельная работа (состояние «р») и работа с памятью (состояние «об»).

В режиме работы с памятью появляется дополнительное состояние ожидания (состояние «ож») из-за наличия второго процессора, который может занимать шину. Условие задачи говорит о том, что при сбое процессор «запоминает» свой режим работы и возвращается в него после восстановления, т.е. на самом деле есть два вида сбоев – сбой в режиме самостоятельной работы и сбой в режиме работы с памятью. При этом важно помнить, что режиму работы с памятью отвечают два состояния: обмен и ожидание.

Формализуем сказанное в виде допущений, а затем, опираясь на различные допущения, построим графы состояний предложенной системы в соответствии с описанной выше методикой.

#### Формирование множества ДФ.

ДФ<sub>1</sub>: Каждый из процессоров выполняет обработку информации независимо друг от друга и обращается к памяти в произвольные (случайные) моменты времени с интервалом  $\tau_1, \tau_2$ , определяемые ходом программы.

ДФ<sub>2</sub>: Обращение к памяти осуществляется через общую шину и не может происходить одновременно. В случае занятости шины процессор переходит в состояние ожидания.

ДФ<sub>3</sub>: Время обмена данными с памятью является случайной величиной  $\tau_{об1}, \tau_{об2}$ .

ДФ<sub>4</sub>: При освобождении шины (после обмена данными одного процессора) ее занятие другим процессором (в случае необходимости обмена) происходит мгновенно.

ДФ<sub>5</sub>: Времена  $\tau_1, \tau_2, \tau_{об1}, \tau_{об2}, \tau_{ож1}, \tau_{ож2}$  распределены по экспоненциальному закону.

Граф состояний системы, в которой учитывается перечень допущений ДФ<sub>1</sub> – ДФ<sub>5</sub>, представлен на рис. 2.

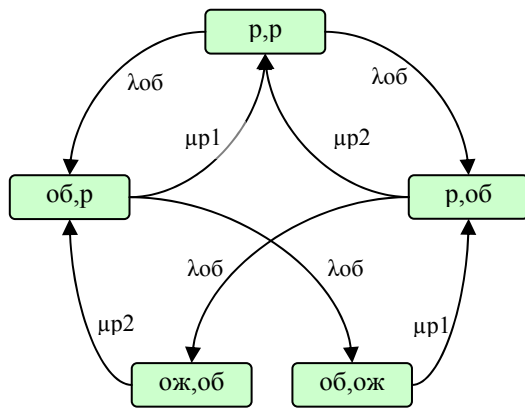


Рис. 2. Граф состояний системы без сбоев (используемые допущения:  $ДФ_1 - ДФ_5$ )

### Формирование множества ДН.

$ДН_{0.1}$ : С точки зрения надежности процессоры могут находиться в трех состояниях: работоспособности (р), сбоя (с) и восстановления.

$ДН_{0.2}$ : Восстановление осуществляется с помощью специальных программ в течение времени  $\tau_{e1}, \tau_{e2}$ .

$ДН_{0.3}$ : Сбой процессора возможен как в состоянии обработки, так и при обращении к памяти (в состоянии обмена или ожидания). После восстановления процессор возвращается к выполнению прерванной операции.

Далее даны допущения и приведены графы состояний для исследуемой системы в зависимости от сбоев процессоров.

### Система с возможностью сбоя только первого процессора.

$ДН_{1.1}$ : В системе возможен сбой только первого процессора.

Граф состояний системы, в которой учитывается перечень допущений  $ДФ_1 - ДФ_5, ДН_{0.1} - ДН_{0.3}$ ,  $ДН_{1.1}$ , представлен на рис. 3.

### Система с возможностью сбоя только второго процессора.

$ДН_{2.1}$ : В системе возможен сбой только второго процессора.

Граф состояний системы, в которой учитывается перечень допущений  $ДФ_1 - ДФ_5, ДН_{0.1} - ДН_{0.3}$ ,  $ДН_{1.2}$ , представлен на рис. 4.

### Система с поочередным нахождением в состоянии сбоя обоих процессоров.

$ДН_{3.1}$ : В системе возможен сбой обоих процессоров.

$ДН_{3.2}$ : Оба процессора не могут одновременно находиться в состоянии сбоя.

Граф состояний системы, в которой учитывается перечень допущений  $ДФ_1 - ДФ_5, ДН_{0.1} - ДН_{0.3}$ ,

$ДН_{3.1} - ДН_{3.2}$ , представлен на рис. 5.

### Система с поочередным сбоем и возможностью одновременного нахождения в состоянии сбоя обоих процессоров.

$ДН_{4.1}$ : В системе возможен сбой обоих процессоров

$ДН_{4.2}$ : Оба процессора могут одновременно находиться в состоянии сбоя.

$ДН_{4.3}$ : Оба процессора не могут одновременно перейти из работоспособного состояния (обработки или обращения к памяти) в состояние сбоя.

Граф состояний системы, в которой учитывается перечень допущений  $ДФ_1 - ДФ_5, ДН_{0.1} - ДН_{0.3}$ ,

$ДН_{4.1} - ДН_{4.3}$ , представлен на рис. 6.

### Разметка графов.

При обозначении вершин для наглядности использован буквенный код ( $X_1, X_2, X_3$ ), где  $X_i$  – состояние  $i$ -го процессора;  $X_1, X_2$  и  $X_3$  могут принимать следующие значения:

«р», «об», «ож», «с1» – сбой и восстановление в режиме самостоятельной работы;

«с2» – сбой и восстановление в режиме обмена с памятью.

Переходы имеют следующие интенсивности:  $\lambda_{об1}, \lambda_{об2}$  – интенсивность перехода в режим обмена;  $\mu_{р1}, \mu_{р2}$  – интенсивность перехода в режим самостоятельной работы;  $\lambda_{с}$  – интенсивность сбоя;  $\mu_{в}$  – интенсивность восстановления.

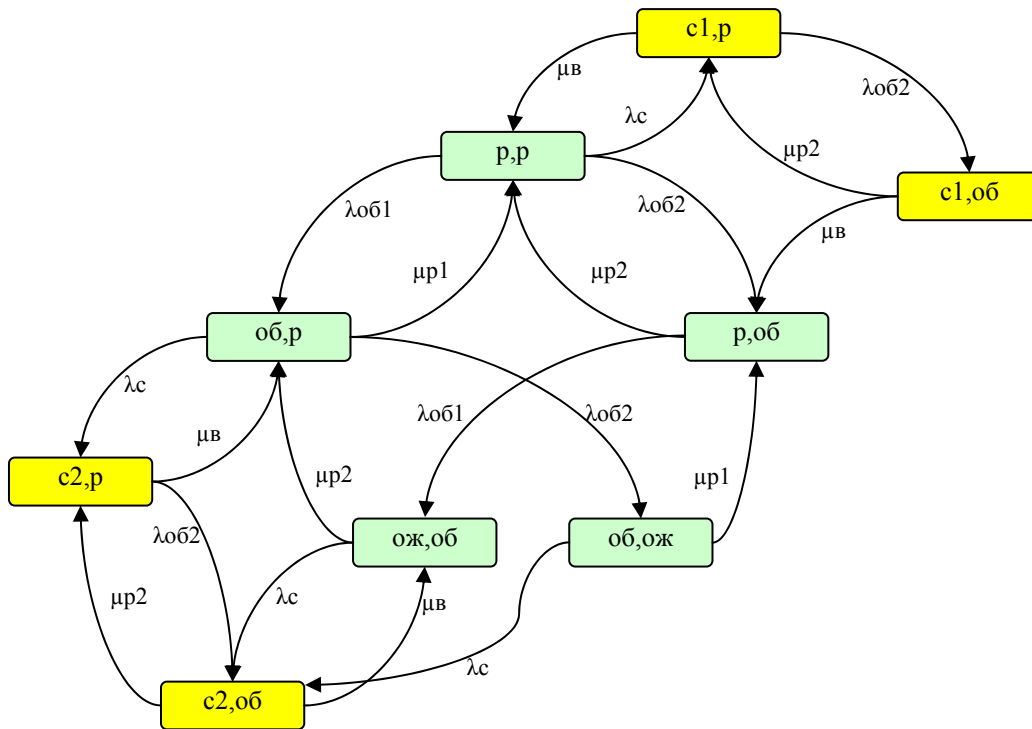


Рис. 3. Граф состояний системи со сбоем первого процессора (используемые допущения:  $ДФ_1 - ДФ_5$ ,  $ДН_{0.1} - ДН_{0.3}$ ,  $ДН_{1.1}$ )

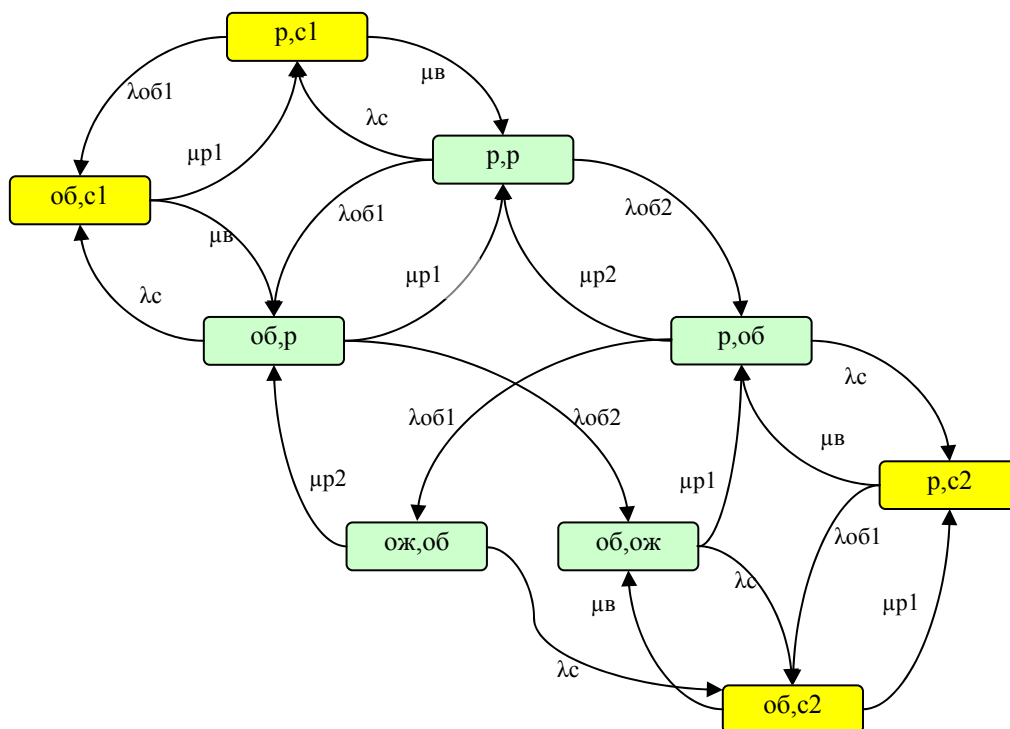


Рис. 4. Граф состояний системы со сбоем второго процессора. (используемые допущения:  $ДФ_1 - ДФ_5$ ,  $ДН_{0.1} - ДН_{0.3}$ ,  $ДН_{1.2}$ )

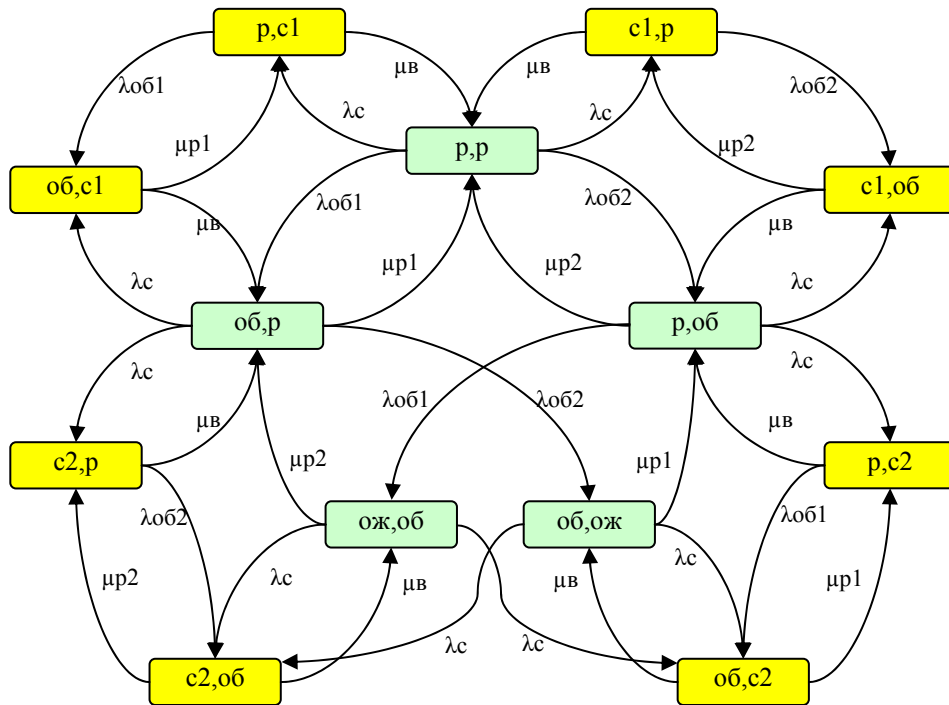


Рис. 5. Граф состояний системи со сбоем двух процессоров.  
(используемые допущения:  $ДФ_1 - ДФ_5$ ,  $ДН_{0,1} - ДН_{0,3}$ ,  $ДН_{3,1} - ДН_{3,2}$ )

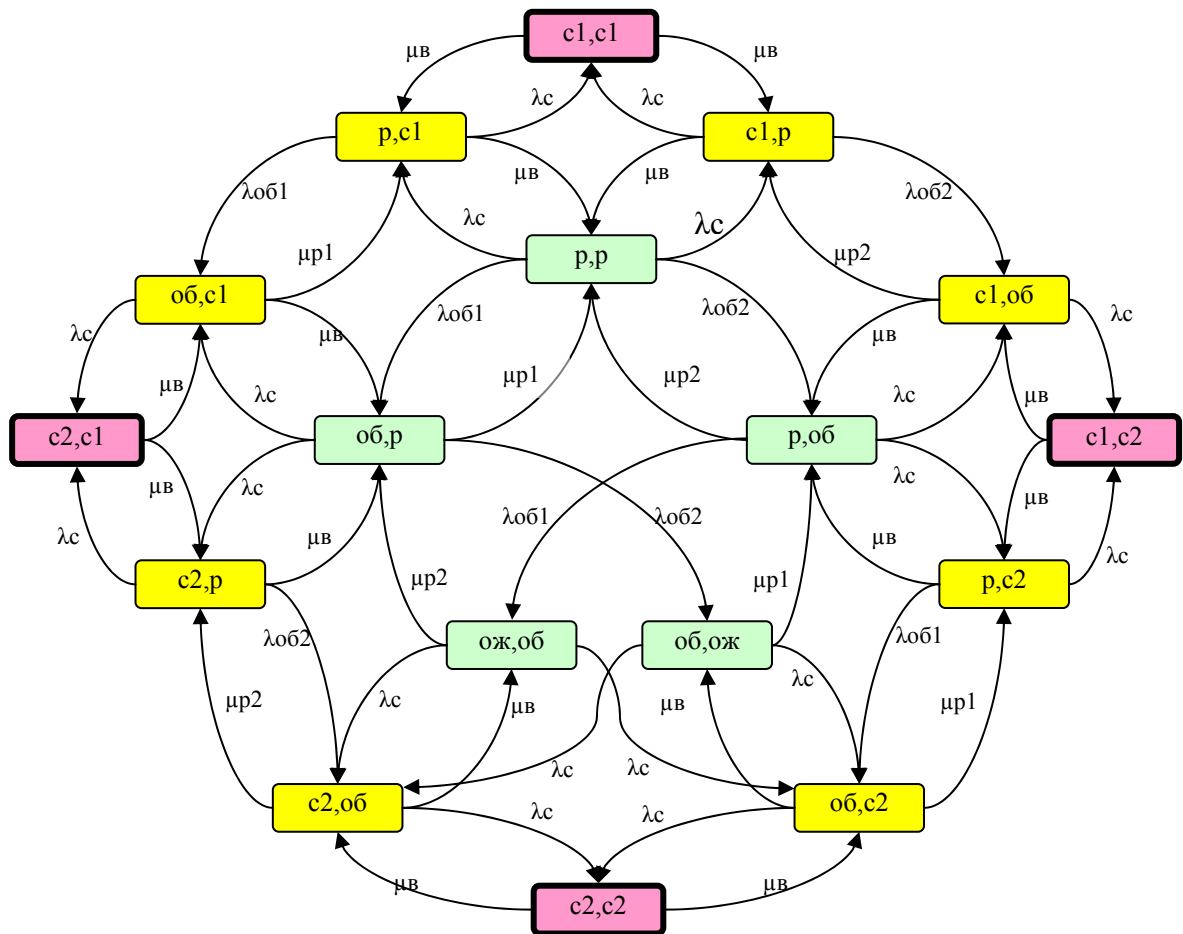


Рис. 6. Граф состояний системы со сбоем двух процессоров.  
(используемые допущения:  $ДФ_1 - ДФ_5$ ,  $ДН_{0,1} - ДН_{0,3}$ ,  $ДН_{4,1} - ДН_{4,3}$ )

Анализ полученных графов состояний (рис. 2 – 6) показывает, что поэтапное расширение множества допущений, приводит к поэтапному расширению множества состояний и соответствующих вершин (например, при введении допущений, связанных с возможностью возникновения сбоев процессоров появляются дополнительные состояния:

– рис. 3 ( $c_{1.p}$ ), ( $c_{2.p}$ ), ( $c_{1.об}$ ), ( $c_{2.об}$ );

– рис. 4 ( $p.c_1$ ), ( $p.c_2$ ), ( $об.c_1$ ), ( $об.c_2$ );

– граф состояний рис. 5 является объединением графов состояний рис. 3 и 4.

Аналогично, на следующем шаге при построении графа состояний рис. 6 появляются дополнительные состояния, учитывающие возможность возникновения сбоя одного процессора до восстановления работоспособности после сбоя другого процессора (рис. 6 – ( $c_{1.c_1}$ ), ( $c_{1.c_2}$ ), ( $c_{2.c_1}$ ), ( $c_{2.c_2}$ )).

### Заключение

Проведенный анализ задач оценки и исследования характеристик производительности и надежности вычислительных систем с использованием аппарата теории марковских процессов и стандартных средств компьютерного моделирования показывает, что достоверность и точность результатов исследования определяется, в первую очередь, корректностью и полнотой построения и разметки графа состояний (матрицы интенсивностей переходов).

Предложенная методика построения марковской модели позволяет обеспечить трассируемость допущений о функционировании системы, а также упрощает решение проблемы адекватности модели реальной ВС на основе пошагового учета (снятия) ограничивающих допущений, определяемых режимами функционирования и обеспечения надежности.

Методика проиллюстрирована на примере построения марковского графа для двухпроцессорной вычислительной системы.

### Литература

1. Романкевич А.М., Карачун Л.Ф., Романкевич В.А. Графо-логические модели для анализа сложных отказоустойчивых ВС // Электронное моделирование. – 2001. – Т. 23, № 1. – С. 102 – 111.
2. Гультьев А. Визуальное моделирование в среде MATLAB. – СПб: Питер, 2000. – 432 с.
3. Креденцер Б.П., Сидоров Л.А., Буточнов А.Н. Применение операции укрупнения состояний для оценки надежности одного класса резервированных систем // Кибернетика. – 1990. – № 2. – С. 26 – 29.
4. Черкесов Г.Н. Надёжность программно-аппаратных комплексов. – С-Пб.: Питер, 2004. – 472 с.
5. Основы цифровых систем: Підручник / За ред. М.П. Благодарного, В.С. Харченка. – Х.: Міносвіти України, 2002. – 548 с.
6. Одарущенко О.Н., Одарущенко Е.Б., Поночевный Ю.Л. Применение численных методов для решения жестких систем линейных дифференциальных уравнений в задачах оценки надежности // Авіаційно-космічна техніка і технологія. – Х.: Нац. аерокосм. ун-т „ХАІ”, 2002. – Вип. 35. – С. 187 – 191.
7. Скляр В.В., Харченко В.С. Отказоустойчивые компьютерные системы управления с версионно-пороговой адаптацией // Автоматика и телемеханика. – 2002. – № 6. – С. 131 – 145.
8. Харченко В.С., Гридин Ю.В. Модели бортовых вычислительных систем с категорированием задач в условиях сбоев и отказов аппаратных и программных средств // Космічна наука і технологія. – 1999. – Т. 5, № 5/6. – С. 101 – 103.
9. Харченко В.С., Асидех Ф.А., Лысенко И.В. Марковские модели готовности восстанавливаемых STRATUS-систем // Системи обробки інформації. – Х.: ХВУ. – 2004. – Вип. 4. – С. 216 – 226.

*Поступила в редакцию 15.07.2005*

**Рецензент:** д-р техн. наук, проф. В.А. Краснобаев, Харьковский национальный технический университет сельского хозяйства, Харьков.