УДК 519.8

**Б.В. ГРЕЧУК, М.Г. ФУРУГЯН, В.А. ПЕТРУХИН**

# ОПТИМАЛЬНЫЕ СТРАТЕГИИ ОРГАНИЗАЦИИ РЕСТАРТОВ В СИСТЕМАХ РЕАЛЬНОГО ВРЕМЕНИ

***Abstract:*** *The problem of restart organization in real-time systems is considered. The importance of the order of performance of working and supervising modules is shown. An alternative strategy of error handling, according to which the restart of the system does not follow immediately after error detection by a verifying module is proposed. An example according to which new proposed strategy leads to reduction of the expected execution time is considered.*
***Key words:*** *real-time system, computations verification, restart organization, checkpointing.*

***Анотація:*** *Розглядається задача організації перезапусків в системах реального часу. Показана важливість порядку виконання робочих та контролюючих модулів. Запропонована альтернативна стратегія поведінки при виявленні помилки, при якій перезапуск системи відбувається не відразу після виявлення помилки деяким контролюючим модулем. Розглянуто приклад, при якому нова стратегія призводить до зменшення середнього часу роботи системи.*
***Ключові слова:*** *система реального часу, підсистема контролю даних, організація перезапусків.*

***Аннотация:*** *Рассматривается задача организации рестартов в системах реального времени. Показана важность порядка выполнения рабочих и контролирующих модулей. Предложена альтернативная стратегия поведения при обнаружении ошибки, при которой рестарт системы происходит не сразу после обнаружения ошибки некоторым модулем контроля. Рассмотрен пример, в котором новая стратегия приводит к уменьшению среднего времени работы системы.*
***Ключевые слова:*** *система реального времени, подсистема контроля данных, организация рестартов.*

## 1. Введение

При планировании вычислений в системах реального времени важно учитывать возможность несистематического возникновения программных сбоев и ошибок. Если система не защищена от программных сбоев, то при их возникновении все вычисления необходимо каждый раз повторять заново, что часто требует больших затрат времени и ресурсов. Стандартным способом защиты от сбоев и ошибок является введение в систему дополнительных модулей-буферов, которые сохраняют промежуточную информацию о состоянии системы. Эти модули используются для того, чтобы при рестарте воспользоваться сохраненными данными. Таким образом, наличие модулей-буферов может существенно улучшить надежность системы и уменьшить временные задержки, вызванные несистематическими сбоями и ошибками. С другой стороны, избыточное использование модулей-буферов может привести к существенному уменьшению быстродействия и увеличению стоимости системы. Следовательно, возникает задача оптимальной частоты расположения модулей-буферов. В работах [1–5] предполагается, что при возникновении ошибки мы сразу же можем ее обнаружить и выполнить перезапуск системы. На практике часто возникают ошибки, которые невозможно обнаружить сразу. В этом случае контроль системы осуществляется специальными модулями контроля, каждый из которых определяет, принадлежит ли множество контролируемых им параметров заданным пределам. Тогда перезапуск системы происходит не сразу после возникновении ошибки, а только после ее обнаружения некоторым модулем контроля. Подсистема контроля данных подробно рассмотрена в [6].

В работе [7] построена общая модель организации рестартов в системах реального времени, включающая в себя как буферизацию, так и контроль данных. Вычислительная система реального времени рассматривается как граф, вершинами которого являются рабочие модули, модули контроля и модули-буфера, а ребра обозначают зависимости по данным. Ранее была рассмотрена задача оптимального расположения модулей контроля и модулей-буферов [8, 9]. Все

полученные результаты основывались на логичном предположении, что перезапуск системы автоматически происходит сразу же при обнаружении ошибки некоторым модулем контроля.

Цель настоящей статьи – рассмотреть новую стратегию организации рестартов в системах реального времени. Будет показано, что в некоторых случаях имеет смысл после обнаружения ошибки продолжать работу системы, не делая перезапуска сразу же. Эта стратегия может позволить уменьшить математическое ожидание времени работы системы, не увеличивая количество модулей контроля и модулей-буферов, и даже не меняя их расположения. Значение имеет также и порядок выполнения модулей контроля.

**2. Модель организации рестартов**

Остановимся более подробно на модели организации рестартов в системах реального времени, рассмотренной в [7]. Дан ориентированный граф без циклов, вершины которого программные модули, а ребра – зависимости по данным между ними (ребро $(i, j)$ означает, что модуль $i$ передает данные модулю $j$). Модули делятся на 3 типа:

– рабочие модули $A_i$, $i = 1,2,...,n$. Это главные модули, они выполняют вычисления. Время выполнения каждого рабочего модуля предполагается одинаковым и равным единице. При выполнении каждого рабочего модуля с вероятностью $a > 0$ может возникнуть ошибка и тогда его необходимо выполнить повторно;

– модули контроля $C_i$, $i = 1,2,...,k$, которые проверяют рабочие модули на наличие ошибок. Предполагается, что время работы каждого такого модуля равно нулю, и он указывает на ошибку в том и только том случае, если хотя бы в одном из предшествующих ему по графу модулей произошла ошибка;

– модули-буфера $B_i$, $i = 1,2,...,m$, которые сохраняют полученные данные и передают их последующим модулям. Предполагается, что время работы каждого такого модуля также равно нулю. Для повторного выполнения рабочего модуля $A_i$ нужно повторно выполнить все рабочие модули, которые непосредственно ему предшествуют, чтобы получить данные для $A_i$. Для каждого из его предшественников, в свою очередь, необходимо повторно выполнить рабочие модули, предшествующие ему, и т.д., пока не дойдем до модулей-буферов.

В [7] решается задача построения минимального множества работ, которые необходимо повторно выполнить при обнаружении ошибки некоторым модулем контроля. В [8, 9] рассмотрена задача расположения модулей контроля и модулей-буферов так, чтобы математическое ожидание суммарного времени, затраченного на выполнение всех модулей (включая повторное их выполнение при возникновении ошибок), было минимальным. Минимум ищется по всем допустимым расстановкам. Расстановка называется допустимой, если ошибка в любом рабочем модуле может быть обнаружена некоторым модулем контроля и все данные могут быть восстановлены. Иными словами, для каждого рабочего модуля существует ориентированный путь, ведущий из этого модуля в некоторый модуль контроля, и ориентированный путь, ведущий из некоторого модуля-буфера в этот рабочий модуль.

В работах [7–9] предполагается, что перезапуск системы происходит сразу же при обнаружении ошибки некоторым модулем контроля. В дальнейшем такую стратегию наших действий при обнаружении ошибки будем называть *простой стратегией*. В следующих разделах мы рассмотрим как простую, так и альтернативную стратегии организации рестартов в системах реального времени.

### 3. Порядок выполнения модулей контроля при простой стратегии

В этом и следующем разделах для простоты предположим, что вероятность возникновения ошибки при выполнении каждого рабочего модуля $a << \frac{1}{n}$, а перед каждым рабочим модулем расположен модуль-буфер [8, 9].

Далее покажем, что даже при фиксированной расстановке модулей контроля и простой стратегии наших действий математическое ожидание времени работы всей системы может быть различным в зависимости от порядка выполнения рабочих модулей и модулей контроля.

Рассмотрим простой пример, когда вершинами графа $G$ являются 5 рабочих модулей $A_1, A_2, ..., A_5$ (вместе с модулями-буферами перед каждым из них), а ребрами – 4 зависимости по данным $(A_1, A_2)$, $(A_2, A_3)$, $(A_3, A_4)$ и $(A_2, A_5)$ (рис. 1). Также имеются два модуля контроля $C_1$ и $C_2$. Тогда единственная допустимая расстановка состоит в том, чтобы расположить модули контроля $C_1$ и $C_2$ после рабочих модулей $A_4$ и $A_5$. Заметим, что так как $a << \frac{1}{n}$, то вероятность возникновения более чем одной ошибки пренебрежимо мала и даже не рассматривается.

Рассмотрим следующие варианты задания порядка выполнения рабочих модулей и модулей контроля.

1. Выполняем модули $A_1, A_2, A_3, A_4, C_1$. Если модуль контроля $C_1$ обнаружил ошибку, повторяем выполнение этих модулей, пока они не выполнятся без ошибки. После этого выполняем $A_5$ и $C_2$. При ошибке повторяем выполнение $A_5$ и $C_2$.

Рис. 1. Граф зависимости по данным. Пример 1

2. Выполняем модули $A_1, A_2, A_5, C_2$. Если модуль контроля $C_2$ обнаружил ошибку, повторяем выполнение этих модулей, пока они не выполнятся без ошибки. После этого выполняем $A_3, A_4, C_1$, пока $C_1$ не выполнится без ошибки.

Длительности выполнения рабочих модулей равны 1, модулей контроля – 0. Вероятность возникновения ошибки в каждом из рабочих модулей равна $a$. Пренебрегая малыми порядка $a^2$, получим, что при первом порядке выполнения модулей математическое ожидание общего потраченного времени

$$M_1 = (1 - 5a) \cdot 5 + 4a(5 + 4) + a(5 + 1) = 5 + 17a.$$

При втором порядке выполнения модулей аналогично получаем

$$M_2 = (1-5a)\cdot 5 + 3a(5+3) + 2a(5+2) = 5 + 13a\,.$$

То есть при изменении порядка выполнения модулей контроля результат изменился. Объяснением данного результата является то, что один и тот же рабочий модуль может контролироваться несколькими независимыми модулями контроля.

**4. Альтернативная стратегия при обнаружении ошибки**

В предыдущем разделе, как и в работах [7–9], предполагалось, что при обнаружении ошибки модулем контроля рестарт системы происходит немедленно. В этом разделе мы рассмотрим пример, в котором проиллюстрируем, что такая стратегия не всегда является оптимальной.

Представим систему из 9 рабочих модулей $A_1, A_2, ..., A_9$ с зависимостями по данным $(A_1, A_2)$, $(A_3, A_4)$, $(A_5, A_6)$, $(A_2, A_7)$, $(A_4, A_7)$, $(A_2, A_8)$, $(A_6, A_8)$, $(A_4, A_9)$, $(A_6, A_9)$ (рис. 2). Также имеются три модуля контроля $C_1, C_2, C_3$. Тогда единственная допустимая расстановка состоит в том, чтобы расположить модули контроля $C_1, C_2, C_3$ после рабочих модулей $A_7, A_8, A_9$.

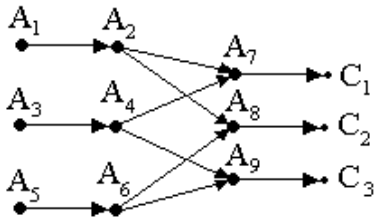

Рис. 2. Граф зависимости по данным. Пример 2

Из симметрии следует, что для этого графа от порядка выполнения модулей контроля ничего не зависит, и все простые стратегии эквивалентны.

Рассмотрим, например, следующий порядок.

1. Выполняем модули $A_1, A_2, A_3, A_4, A_7, C_1$, при ошибке повторяем их. Потом выполняем $A_5, A_6, A_8, C_2$, при ошибке повторяем их. И, наконец, $A_9$ и $C_3$, при ошибке повторяем их. Напомним, что из условия $a << \frac{1}{n}$ следует, что возможность возникновения более чем одной ошибки можно не учитывать, а малыми порядка $a^2$ можно пренебречь. Тогда для математического ожидания времени работы системы получаем

$$M_1 = (1-9a)\cdot 9 + 5a(9+5) + 3a(9+3) + a(9+1) = 9 + 35a\,.$$

Рассмотрим альтернативную стратегию, не являющуюся простой.

2. Выполняем все модули до конца, не обращая внимания на ошибки. Если все модули контроля показывают ошибки, повторяем все сначала (впрочем, для такого случая нужны ошибки одновременно как минимум в двух рабочих модулях, а вероятность этого пренебрежимо мала). Если ошибки показывают модули $C_1$ и $C_2$ (но не $C_3$), повторяем выполнение рабочих модулей $A_1, A_2, A_7, A_8$ (т.е. тех, которые не контролирует $C_3$). Этот случай реализуется при ошибке в модулях $A_1$ и $A_2$. Аналогично, если ошибки показывают модули $C_1$ и $C_3$, повторяем выполнение $A_3, A_4, A_7, A_9$, а если модули $C_2$ и $C_3$, то повторяем $A_5, A_6, A_8, A_9$. Если же ошибку показал модуль $C_1$, надо повторить только рабочий модуль $A_7$ (ошибка определенно в нем). Аналогично, если об ошибке сигнализирует $C_2$, надо повторить $A_8$, а если $C_3$ – повторяем $A_9$.

При такой стратегии для математического ожидания работы всей системы получаем

$$M_2 = (1-9a)\cdot 9 + 3\cdot 2a(9+4) + 3\cdot a(9+1) = 9 + 27a < M_1.$$

Как видим, в этом примере более выгодно при обнаружении ошибки не приступать к перезапуску системы немедленно. Более выгодно сначала выполнить все рабочие модули и модули контроля и только после этого выполнить рестарт системы.

**5. Заключение**

В статье рассмотрена альтернативная стратегия поведения при обнаружении ошибки в системах реального времени. Показано, что стратегия, когда при обнаружении ошибки модулем контроля рестарт системы происходит немедленно, не всегда является оптимальной. Приведены примеры, в которых более эффективным оказывается метод отложенного перезапуска системы. Этот результат может быть использован для уменьшения математического ожидания времени работы системы реального времени без увеличения количества модулей контроля и модулей-буферов или изменения их расположения.

Идея отказа от немедленного перезапуска системы при обнаружении ошибки является новой в рамках данной модели организации рестартов в системах реального времени. На основании данной идеи могут быть разработаны новые алгоритмы расположения модулей контроля и модулей-буферов, при которых среднее время работы системы существенно уменьшится по сравнению со стандартным подходом, когда перезапуск системы происходит немедленно после обнаружения ошибки. Кроме того, метод отложенного перезапуска может быть использован и в некоторых других моделях организации рестартов. На практике это приведет к повышению надежности и эффективности систем реального времени без существенного увеличения их стоимости.

**СПИСОК ЛИТЕРАТУРЫ**

1. Young J. W. A first-order approximation to the optimum checkpoint interval // Comm. ACM. – 1974. – 17, 9 – P. 530–531.
2. Gelembe E. On the Optimum Checkpoint Interval // J. ACM. – 1979. – Vol. 26. – P. 259–270.
3. Coffman E., Gilbert E. Optimal Strategies for Scheduling Checkpoints and Preventive Maintenance // IEEE Trans. Reliability. – 1990. – Vol. 39, N 1. – P. 9–18.
4. Tantawi A., Ruschitzka M. Performance Analysis of Checkpointing Strategies // ACM Trans. Computer Systems. – 1984. – Vol. 2, N 2. – P.123–144.
5. Ozaki Tatsuya, Dohi Tadashi, Okamura Hiroyuki, Kaio Naoto Min-Max Checkpoint Placement under Incomplete Failure Information // International Conference on DSN. – 2004.
6. Луганская М.И., Сушков Б.Г. Контроль данных в системах реального времени // Математические методы управления обработкой информации. – М.: МФТИ, 1986. – С. 18–24.
7. Белый Д.В., Сушков Б.Г. Модель организации рестартов в системах реального времени. – М.: ВЦ РАН, 1996.
8. Гречук Б.В., Фуругян М.Г. Алгоритмы организации рестартов в системах реального времени. – М.: ВЦ РАН, 2004. – С. 234–237.
9. Гречук Б.В., Фуругян М.Г. Алгоритмы организации рестартов в системах реального времени с произвольным графом связей. – М.: ВЦ РАН, 2005. – 34 с.

*Стаття надійшла до редакції 01.03.2006*