Отказоустойчивые ВС

Майданов
Юрий Сергеевич
к.т.н., доцент Кафедры ВС

Ускоренная дешифрация синдрома вычислительной системы

Функционирование в условиях реального времени налагает жесткие временные ограничения на работу вычислительной системы. Поэтому актуальна задача получения алгоритмов работы таких систем, оптимальных в конкретных условиях (возможно в ущерб унификации). Для решения задачи дешифрации синдрома вычислительной системы такие алгоритмы могут быть получены путем учета частных свойств используемой диагностической модели

Так как для любой диагностической модели синдром полностью исправной системы (все элементарные машины исправны) состоит только из нулевых элементов, то дешифрацию синдрома системы не имеет смысла проводить до тех пор, пока он не содержит хотя бы одного единичного элемента

Ускоренная дешифрация для модели (0,1,0,0)

Для этой модели характерно получение единичного результата элементарной проверки только в одном случае – когда контролирующая элементарная машина исправна, а контролируемая вышла из строя

Для определения состояния вычислительной системы достаточно выделить множество вершин диагностического графа, для каждой из которых существует хотя бы одна входящая дуга взвешенная единицей. Элементарные машины, соответствующие таким вершинам неисправны

Алгоритм для модели (0,1,0,0)

1. Пометить все машины в составе вычислительной системе как исправные.

2. Выбрать элемент из синдрома системы s_{ji}.

3. Если выбранный элемент равен «0», то выполнить переход к п. 5.

4. Контролируемая машина u_i помечается как неисправная.

5. Если выбраны не все элементы синдрома, то выполнить переход к п.2.

6. Конец алгоритма.

Свойства алгоритма для модели (0,1,0,0)

Порядок выбора элементов синдрома ничем не ограничивается. Поэтому предложенный алгоритм может выполняться по мере сбора диагностической информации (т.е. по мере формирования синдрома системы)

Для выделения неисправных элементарных машин может использоваться частично заполненный синдром вычислительной системы, что обуславливается спецификой данной диагностической модели

Все это дает возможность эффективно использовать его в системах реального времени

Свойства алгоритма для модели (0,1,0,0)

Алгоритм реализует один цикл просмотра синдрома системы, после чего выносится решение о техническом состоянии. Поэтому трудоемкость дешифрации пропорциональна количеству связей m диагностического графа

$$m \leq n\,(n-1) < n^2$$

Таким образом, вычислительная сложность определяется:

$$O(\,n^2\,)$$

Минимальны и затраты памяти, так как необходимо хранить только состояние системы

Ускоренная дешифрация для модели (0,1,0,1)

Эту диагностическую модель называют идеальной, потому что состояние контролируемой элементарной машины однозначно правильно определяется независимо от состояния контролирующей

Аналогично диагностической модели (0,1,0,0) выделение множества неисправных машин сводится к поиску тех вершин, в которые входят дуги взвешены в синдроме единичным значением

Поэтому для дешифрации синдрома вычислительной системы может быть использован алгоритм от модели (0,1,0,0)

Ускоренная дешифрация для модели (0,1,0,X)

Обладает теми же особенностями, что и (0,1,0,1) за исключением того, что не гарантировано верное определение состояния машины неисправной контролирующей машины

В худшем случае в текущем цикле диагностирования все элементарные проверки, в которых в качестве контролирующих выступают неисправные машины, всегда имеют результат 0. При этом работа вычислительной системы может быть описана моделью (0,1,0,0). Если результатом таких элементарных проверок является 1, то будет иметь место идеальная диагностическая модель. Как в первом, так и во втором случае, используя алгоритм модели (0,1,0,0), можно определить техническое состояние вычислительной системы

Ускоренная дешифрация для модели (0,1,1,0)

Для этой модели характерно получение единичного результата тех элементарных проверок, в которых одна из машин исправна, а другая вышла из строя

Необходимым и достаточным условием t – диагностируемости вычислительной системы является связанность диагностического графа. Поэтому перед работой алгоритма следует из диагностического графа исключить те ребра, без которых он остается связанным, или не рассматривать их при дешифрации. При этом возможно применение алгоритмов Прима и Краскала

Алгоритм для модели (0,1,1,0)

Сначала строится граф $G_0 = (U, T_0)$, который является подграфом исходного диагностического графа $G = (U, T)$ и образуется путем удаления дуг (u_i, u_j), взвешенных единичным значением $s_{ij} = 1$. Граф G_0 состоит из таких компонент связности, где состояние вершин, входящих в одну компоненту, одинаково. Каждая компонента связности G_{0i} в графе G “окружена” компонентами связности противоположного состояния. Если принять, что G_{0i} содержит исправные вершины, то все компоненты, составляющие ее окружение, содержат неисправные вершины, а последние в свою очередь окружены компонентами, состоящими из исправных вершин, и так далее для каждого очередного «слоя» окружения.

В графе G_0 определяется количество вершин в четных слоях n_1. Если $n_1 < t$, то вершины четных слоев – неисправные, а нечетных – исправные. Если $n_1 > t$, то наоборот вершины четных слоев – исправные, а нечетных – неисправные.

Процедура получения графа G_0 выполняется за один просмотр связей исходного диагностического графа. Остальные процедуры также обладают малой трудоемкостью.

Ускоренная дешифрация для модели (0,1,1,1)

Если участвующие во взаимопроверке элементарные машины исправны, то ее результатом будет «0», во всех остальных случаях — «1»

Выделим множество вершин $U_1 = \{ u_i \}$, $U_1 \subset U$, для каждой вершины которого имеется хотя бы одно ребро диагностического графа $s_{ij} = 0$.

Очевидно, что все элементарные машины, составляющие U_1, исправны, так как данная диагностическая модель не допускает нулевого значения результата элементарной проверки при неисправном состоянии одной из участвующих в ней машин.

Ускоренная дешифрация для модели (0,1,1,1)

Определим множество вершин $U_2 = \{ u_j \}$, $U_2 \subset U$, такое, что $\Gamma^{-1}(U_2) = U_1$. Если находится $s_{ix} = 0$ и $\Gamma^{-1}(U_2) = U_1$, то для каждого $u_j \in U_2$ существует $s_{ij} =1$, так как $u_i \in U_1$. Таким образом, для каждого $u_j \in U_2$ существуют элементарные проверки, в которых одна из машин исправна ($u_i \in U_1$), с единичным результатом ($s_{ij} = 1$). В рамках используемой диагностической модели единичный результат элементарной проверки указывает на неисправность одной из машин. Множество U_1 состоит только из исправных машин. Поэтому все машины, составляющие U_2, неисправны.

Ускоренная дешифрация для модели (0,1,1,1)

Пусть $U_3 = (U \setminus U_1) \setminus U_2$. Покажем, что все машины $u_k \in U_3$ исправны. Допустим противоположное. Пусть существует неисправная машина $u_b \in U_3$. Так как $u_i \in U_1$, если находится $s_{ix} = 0$, то для каждого $u_k \in U_3$ не существует s_k=0 независимо от состояния u_k. Тогда синдромы вычислительной системы, соответствующие исправному и неисправному состояниям u_b, совпадут, а это может быть только в том случае, если количество неисправных машин в системе превышает степень диагностируемости t. Видно противоречие, значит исходное предположение о наличии неисправной машины $u_b \in U_3$ не верно. Следовательно, все элементарные машины $u_k \in U_3$ исправны, что и требовалось показать.

Характеристики алгоритма для модели (0,1,1,1)

Для каждого элемента синдрома производится не более одного изменения логического выражения $F(b_1, b_2, ..., b_n)$. При этом размер выражения не превышает одного терма. Каждый элемент синдрома рассматривается один раз. На последнем этапе работы алгоритма выполняется дополнение выражения переменными без инверсии в виде n циклов. Синдром не может содержать более $m = n^2$ элементов. Таким образом, трудоемкость алгоритма может быть оценена, как $O(n^2)$.

Для реализации предложенного выше метода необходим минимальный объем используемой памяти (n ячеек, так как выражение $F(b_1, b_2, ..., b_n)$ состоит только из одного терма, включающего в себя столько логических переменных, сколько машин в системе). При этом трудоемкость процедуры дешифрации синдрома не высока и не зависит от меры диагностируемости t. Однако при этом налагается требование полного формирования синдрома системы S перед началом работы алгоритма. Это приводит к невозможности проводить дешифрацию синдрома по мере поступления данных параллельно с решением основной задачи.

Практическая реализация алгоритма для модели (0,1,1,1)

Чтобы реализовать дешифрацию синдрома вычислительной системы в процессе выполнения элементарных проверок можно использовать следующий подход. Первый этап работы алгоритма (определение множества U_1) проводится по мере поступления данных об элементарных проверках с результатам «0». Если результатом очередной элементарной проверки s_{ij} является «1», но одна из участвующих в ней машин b_i входит в выражение F без инверсии, то оно модифицируется: $F(b_1, b_2, ..., b_n) = F(b_1, b_2, ..., b_n) \wedge \overline{b}_j$. После того, как выполнены все элементарные проверки, определенные диагностическим графом (т. е. синдром системы полностью сформирован), рассматриваются оставшиеся элементы синдрома (они все единичные) и выполняются два последних этапа работы алгоритма.

Множество U_1 определяется в процессе проведения элементарных проверок, $U_2 = \Gamma^{-1}(U_1)$ определяется по диагностическому графу. Аналогично $U_3 = (U \setminus U_1) \setminus U_2$. Хранить синдром S при этом нет необходимости, поскольку все необходимые данные для реализации второго и третьего этапов процедуры дешифрации содержатся в диагностическом графе $G = (U, T)$.

Ускоренная дешифрация при использовании коллективных проверок

В отличие от парных элементарных проверок при проведении коллективных проверок один и тот же фрагмент прикладных задач w_q может решаться более чем на двух элементарных машинах одновременно.

Если в коллективной проверке участвовало r_τ элементарных машин, то по результатам независимого решения фрагмента задачи w_q можно произвести $m_\tau = \dfrac{r_\tau(r_\tau - 1)}{2}$ сравнений. Каждое такое сравнение результатов эквивалентно проведению одной элементарной проверки. Можно построить диагностический граф $G = (U, T)$ и сформировать синдром вычислительной системы, соответствующий множеству проведенных диагностических проверок. Поэтому все рассмотренные методы определения технического состояния вычислительных систем применимы не только при использовании парных элементарных проверок, но и при использовании коллективных проверок.

Ускоренная дешифрация при использовании коллективных проверок

С другой стороны, большое количество парных элементарных проверок, эквивалентных одной коллективной проверке, приводит к увеличению размера синдрома системы и, соответственно, к возрастанию трудоемкости его дешифрации.

Таким образом, применяя коллективные проверки необходимо использовать специальные методы определения технического состояния вычислительной системы, учитывающие особенности выбранного способа проведения взаимопроверок между элементарными машинами.

Так как все участвующие в коллективной проверке элементарные машины решают копии одного и того же фрагмента прикладных задач w_q, то полученные исправными машинами результаты совпадут. Учитывая предположение о t-диагностируемости вычислительной системы можно утверждать, что в ней существует не менее $n - t$ исправных машин, получающих одинаковые результаты в любой коллективной проверке.

Спасибо за внимание!