

Для получения значений $A(z)$ в системе класса вычетов с основаниями $p_1(z), p_2(z), \dots, p_n(z)$ необходимо получить в этой системе значения $a_i(z) \cdot z^l \bmod p_i(z)$. В этом случае остаток по модулю $p_i(z)$ определяется

$$a_i(z) = \left\lfloor \sum_{l=0}^k (a_l^i \cdot z^l) \bmod p_i(z) \right\rfloor_2, \quad (5)$$

где $a_l^i = a_l \bmod p_i(z)$, $i = 1, 2, 3, \dots, n$.

В соответствии с (5), перевод $A(z)$ из ПСС в непозиционную можно свести к суммированию по

$$\begin{aligned} z^{12} &\equiv z + 1 \bmod (z^4 + z^3 + 1) \\ z^7 &\equiv z^2 + z + 1 \bmod (z^4 + z^3 + 1) \\ z^5 &\equiv z^3 + z + 1 \bmod (z^4 + z^3 + 1) \end{aligned}$$

Тогда, согласно (5), получаем

$$\begin{aligned} a(z) &= (z+1) \oplus (z^2 + z + 1) \oplus \\ &\oplus (z^3 + z + 1) \oplus (z^3 + 1) \oplus z^3 \oplus z = z^3 + z^2. \end{aligned}$$

Таким образом,

$$\begin{aligned} z^{12} + z^7 + z^5 + z^4 + z^3 + z &\equiv \\ &\equiv z^3 + z^2 \bmod (z^4 + z^3 + 1). \end{aligned}$$

В работе [1] представлена матрица связанности, т. е. синаптические веса нейронной сети, представляются в виде матрицы, строки которой соответствуют области аксонов предыдущего слоя, а столбцы – рецепторным полям нейронов последующего слоя. Разрабатываемая НС для перевода из ПСС в ПСКВ содержит 2 слоя. Первый слой состоит из 15 нейронов, на входы которых подается исходный полином в двоичном коде. С выходов нейронов первого слоя сигналы поступают на входы нейронов 2-го слоя в соответствии с матрицей T_{12}

$$T_{12} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}^T$$

При этом операция перевода осуществляется всего за одну итерацию, что является существенным преимуществом по сравнению с ранее рассмотренными методами перевода. Структура НС, реализующей перевод по $p(z) = z^4 + z^3 + 1$ в ПСКВ поля $GF(2^4)$, представлена [1].

Таким образом, очевидно, что реализация метода непосредственного суммирования для полиномиальной системы класса вычетов позволяет разрабатывать высокоскоростные преобразователи кодов для вычислительных структур реального масштаба времени.

модулю два величин $(a_l^i \cdot z^l) \bmod p_i(z)$ в соответствии с заданным полиномом $A(z)$.

Пример. Определить остаток $A(z) = z^{12} + z^7 + z^5 + z^4 + z^3 + z$ по модулю $p(z) = z^4 + z^3 + 1$.

Для перевода из ПСС в ПСКВ воспользуемся выражением (5). Тогда значения остатков степеней оснований и коэффициентов при них равны

$$\begin{aligned} z^4 &\equiv z^3 + 1 \bmod (z^4 + z^3 + 1) \\ z^3 &\equiv z^3 \bmod (z^4 + z^3 + 1) \\ z &\equiv z \bmod (z^4 + z^3 + 1) \end{aligned}$$

СПИСОК ЛИТЕРАТУРЫ

1. Элементы компьютерной математики и нейроинформатики /Червяков Н.И., Калмыков И.А., Галкина В.А., Щелкунова Ю.О., Шилов А.А.. – М.: ФИЗМАТЛИТ, 2003. – 216 с.
2. Червяков Н.И., Калмыков И.А., Щелкунова Ю.О., Бережной В.В. Математическая модель нейронных сетей для исследования ортогональных преобразований сигналов в расширенных полях Галуа. – Нейрокомпьютеры: разработка и применение. 2003, №6, с.61-68.
3. Калмыков И.А. Математические модели нейросетевых отказоустойчивых вычислительных средств, функционирующих в полиномиальной системе классов вычетов/Под ред. Н.И. Червякова. – М.: ФИЗМАТЛИТ, 2005. – 276 с.

МОДЕЛЬ СРЕДЫ МУЛЬТИВЕРСИОННОГО ИСПОЛНЕНИЯ ПРОГРАММНЫХ МОДУЛЕЙ

Котенок А.В.

Сибирский государственный аэрокосмический университет имени академика М.Ф. Решетнева, Красноярск

Мультиверсионное программирование, начиная с 1977 года, используется там, где требуется выдача точного результата в реальном времени. Причем, в отличие от других методик программной отказоустойчивости, требование к времени выдачи результата является более жестким. Очевидно, что при создании систем с использованием данной концепции будут возникать трудности, в первую очередь, связанные с построением среды, которая будет производить мультиверсионное выполнение программных модулей.

Основой построения среды мультиверсионного исполнения (СМВИ) является проработка ее структуры и связей между элементами, так как от этого зави-

сит не только способность расширяемости и удобство в использовании среды, но и такие параметры как *надежность* и *производительность*.

Пользуясь общими принципами построения систем управления, можно выделить следующие модульные составляющие такой системы:

- программные модули, представляющие версии вычислительного процесса;
- модуль сравнения двух результатов;
- файл проекта;
- исполнительное устройство (ИУ);
- датчики;
- среда мультиверсионного исполнения.

Рассмотрев содержание структуры, перейдем к более сложному вопросу: определение типов связей между модулями системы. Попробуем определить, в каких модулях может произойти непредвиденный отказ, последствия которого нужно предотвратить. Файл проекта, загружается в начале работы, и сохраняется при завершении — отказу тут произойти негде. Отказ ИУ и датчиков в любом случае приведет к отказу всей системы. Модуль сравнения двух результатов содержит довольно простой алгоритм и просто хорошим тестированием этого модуля можно свести вероятность отказа в нем к нулю. Следовательно, связь этих модулей со средой исполнения можно реализовать через статические или динамические библиотеки. Таким образом, единственное «слабое звено» нашей системы – это версии, так как в них производятся сложные вычислительные операции. Здесь возможны несколько вариантов взаимодействия со средой (представлены, по порядку возрастания потребности в памяти).

- Статическая компоновка: из исходных кодов, либо с использованием статических библиотек
- Компоновка с использованием динамических библиотек
- Компоненты выполнены в отдельных исполняемых модулях

Для запуска проекта необходимо запустить все эти модули. Делать это вручную и нецелесообразно, так как это ставит в зависимость производительность среды от оператора. Поэтому следует создать специальные программы-агенты, следящие за запуском всех модулей и перезапускающие их при необходимости. При использовании такой модели для исполнения всех моделей на одном компьютере, агентов можно встроить в саму среду исполнения. А при сетевом распределении модулей встроить агентов в среду очень сложно, так как возникнут трудности с отслеживанием состояния процесса на удаленной машине и его удаленного перезапуска.

Данную модель можно разделить на две, по виду взаимодействия между исполняемыми модулями:

А) через общие участки памяти (например, файлы)

В) через сетевой протокол

С учетом разнообразия моделей можно сделать вывод о нереальности создания действительно универсальной среды мультиверсионного исполнения программных модулей, в частности, из-за противоре-

чивости требований платформенной независимости и максимальной надежности.

Подводя итог, хотелось бы резюмировать ряд требований к исходному коду для того, чтобы приблизить конкретную реализацию среды исполнения к универсальной. А именно:

- использовать открытый исходный код
- все обращения к функциям ОС вынести в отдельный модуль, и сделать его функционирование прозрачным для СМВИ
- использовать язык С или С++

АНАЛИЗ АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ ЭЛЕКТРОННЫМ ДОКУМЕНТООБОРОТОМ СЕРВИСНЫХ ЦЕНТРОВ

Морозов С.А., Манжула В.Г.

Южно-Российский государственный университет экономики и сервиса, Шахты

На сегодняшний день процесс автоматизации документооборота предприятий является одной из актуальных проблем в сфере информационных технологий. Практически любое предприятие имеет особенности, которые отличают его от других аналогичных предприятий. Эти особенности не позволяют создать универсальную комплексную систему, которая подошла бы всем предприятиям.

Массовое появление на российском рынке офисной техники зарубежных производителей повлекло за собой увеличение спроса на ее гарантийное и послегарантийное обслуживание. Сложившееся положение вещей повлекло рынок офисной техники в серьезную конкурентную борьбу – борьбу за потребителя. Очень важную роль в этом процессе играет отбор сервисных центров для создания сервисной сети. Поэтому сервисные компании проходят жесточайший контроль, и одним из критериев оценки является наличие у сервиса автоматизированной системы управления.

Актуальная проблема, связанная с процессом автоматизации сервисных центров – выбор программных средств позволяющих в кратчайшие сроки и с минимальной потерей финансовых средств достигнуть цели автоматизации производства.

Какие программные продукты существуют в мировом информационном сообществе? Какие из них можно применить для автоматизации сервисных центров российского рынка сервисных услуг?

Все продукты для автоматизации сервисных центров, представленные на рынке можно условно разделить на 4 группы:

- 1) зарубежные системы уровня CRM-систем;
- 2) разработки российских фирм (исключая 1С);
- 3) разработки на базе 1С;
- 4) специализированные разработки.

Для оценки программных средств возьмем такие показатели, как функциональность, стоимость поставки, уровень сервиса и стоимость владения.