

МОДЕЛИРОВАНИЕ ЦИФРОВЫХ СХЕМ С НЕИСПРАВНОСТЯМИ НА РАСПРЕДЕЛЁННЫХ СИСТЕМАХ КЛАССА MIMD

Иванов Д.Е., к.т.н., с.н.с., отдел ТУС, Институт Прикладной Математики и Механики, Скобцов Ю.А., д.т.н., проф., кафедра АСУ, Донецкий национальный технический университет.

Предлагается алгоритм распределённого моделирования цифровых схем с неисправностями на системах класса MIMD. Данный алгоритм основывается на модели, использующей один компьютер в качестве ведущего. Рассматривается эффективность и ускорение параллельной реализации. Приведены численные результаты машинных экспериментов.

Введение. Параллельные вычисления на кластерных структурах являются одной из парадигм преодоления нехватки процессорных ресурсов при решении ресурсоёмких задач. Одной из таких задач в технической диагностике является моделирование работы неисправных цифровых устройств [1-3], которое применяется для определения диагностических свойств тестовой последовательности (например, полноты теста).

Принципиально распределенное моделирование с неисправностями можно осуществить двумя путями:

- разбиением теста на части и моделированием каждой такой части на отдельном процессоре системы; данный подход применялся в [4-5];
- разбиением схемы на подсхемы и моделированием подсхемы на отдельном процессоре. При этом необходимо построение протокола взаимодействия всех компонент вычислительного кластера. Такой подход применён в [6-7].

Одним из недостатков второго подхода авторы считают возрастающие накладные расходы на передачу информации необходимой для моделирования (описание подсхем и входные воздействия), построение протокола взаимодействия и балансировки нагрузки, а также синхронизации вычислительных потоков. Поэтому авторы для реализации выбрали первый подход. Заметим также, что в этом случае для схем большой размерности передача информации между процессорами (описание схемы и тестовых наборов) будет сведена к минимуму и выполняться только один раз, и следует ожидать высокой эффективности распределённой реализации алгоритма.

Алгоритм распределённого моделирования цифровых схем с неисправностями. В качестве последовательного алгоритма, предназначенного для распределённой реализации, авторы выбрали разработанный ими ранее алгоритм одиночного параллельного распространения неисправностей. Данный алгоритм показывает хорошие временные характеристики для контрольных схем [8] и подробно описан в [3].

Как отмечалось выше, в качестве базовой выбрана схема с одним ведущим компьютером. Весь вычислительный процесс разбит на несколько потоков: один поток-сервер и несколько (доступных в данный момент) процессов-клиентов. Опишем кратко работу сервера и клиентов. Один из процессоров в кластере выполняет серверную роль: поиск клиентов процесса, ввод описания схемы и тестовой последовательности, разбиение теста на подтесты, передача данных описаний клиентам и получение результатов. Алгоритм работы серверного процесса приведён на рис.1. Способ взаимодействия сервера с клиентами приведён на рис.2. Отметим, что при такой реализации сведены к минимуму и выполняются только один раз следующие элементы взаимодействия:

```
распределённое_моделирование(схема, тест)
{
    число_клиентов = поиск_клиентов();
    если( число_клиентов != 0 )
    {
        ввод_схемы();
        ввод_теста();
        построение_полного_списка_неисправностей();
        разбиение_списка_неисправностей();
        для( i=0 ; i< число_клиентов ; i++ )
        {
            передать_клиенту_i_описание_схемы();
            передать_клиенту_i_часть_списка_неисправностей();
            передать_клиенту_i_тест();
        }
        для( i=0 ; i< число_клиентов ; i++ )
        {
            получить_список_непроверенных_неисправностей();
        }
        формирование_отчёта();
    }
}
```

Рис.1 Алгоритм работы процесса-сервера.

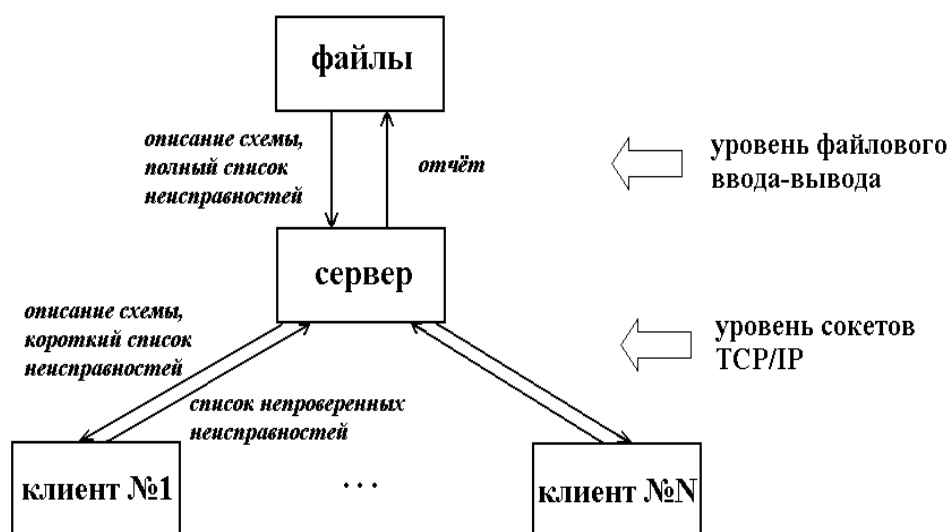


Рис.2 Диаграмма потоков данных при распределённом моделировании.

- передача описания схемы на процессоры-клиенты;
- передача тестовых воздействий на процессоры клиенты;
- синхронизация окончания вычислений.

Каждый процессор-клиент выполняет следующие функции: принимает полное описание схемы, принимает подтест, выполняет моделирование и отправляет его результаты на сервер. Ключевой алгоритм моделирования подробно описан в [1,3].

Программная реализация и анализ. Предложенный распределённый алгоритм реализован программно в среде программирования C++ Builder 6.0. В качестве среды взаимодействия выбраны блокирующие сокеты: процесс, ожидающий приёма данных, будет остановлен до тех пор, пока соответствующий процесс не передаст их.

Машинные эксперименты проводились в компьютерном классе, оборудованном локальной сетью со скоростью 100Мбит/с под управлением операционной системы Windows XP. Проводился замер следующих временных характеристик: общее время процесса моделирования, число событий при моделировании исправной схемы и схем с неисправностями, общее число событий. Для сравнительной базы бралась работа алгоритма авторов [3] на персональном компьютере соответствующей конфигурации. Для анализа была выбрана схема средней размерности S9234.ben [8]. Результаты экспериментов для различного числа процессоров-клиентов приведены в табл.1. В качестве анализа отметим следующие особенности:

Таблица 1. Экспериментальные данные для схемы S9234.ben

Характеристика	Однопроцессорная реализация	Многопроцессорная реализация (число процессоров)					
		1	2	3	4	6	8
длина теста	1000	1000					
общее время моделирования, сек.	330	336	194	138	107	86	79
общее число событий, млн.	440,05	440,05 (1,00)	441,81 (1,00)	443,21 (1,00)	443,78 (1,01)	447,79 (1,01)	449,93 (1,02)
число событий моделирования исправной схемы, млн.	0,48	0,48 (1,00)	0,95 (1,98)	1,42 (2,96)	1,90 (3,96)	2,85 (5,93)	3,80 (7,92)
число событий моделирования схем с неисправностями, млн.	439,58	439,58 (1,00)	440,86 (1,00)	441,79 (1,00)	441,88 (1,00)	444,94 (1,01)	446,13 (1,01)
ускорение	-	0,98	1,7	2,39	3,08	3,83	4,17
эффективность	-	0,98	0,85	0,8	0,77	0,64	0,52

- общее число событий практически не изменяется в зависимости от числа клиентов (коэффициент от 1.00 для числа клиентов 1-4 до 1.02 для числа клиентов 8);

- число событий моделирования исправной схемы растёт практически линейно (коэффициент от 1.00 для числа клиентов 1 до 7.92 для числа клиентов 8), что связано с необходимостью восстанавливать поведение исправной схемы на каждом узле кластера;

- суммарное число событий моделирования неисправной схемы также практически не растёт.

Для параллельной реализации алгоритма принято вычислять параметры ускорение и эффективность реализации [9].

Ускорение, рассчитываемое для параллельной реализации алгоритма для p процессоров определяется формулой:

$$S_p(n) = \frac{T_1(n)}{T_p(n)},$$

где p – число процессоров в параллельной реализации алгоритма, n – параметр вычислительной сложности алгоритма, $T_i(n)$ – время выполнения параллельного алгоритма на системе с i процессорами.

Эффективность использования процессоров при параллельной реализации алгоритма рассчитывается по формуле:

$$E_p(n) = \frac{T_1(n)}{p \cdot T_p(n)} = \frac{S_p(n)}{p}.$$

Имея экспериментальные данные, из параллельного закона Амдаля можно получить f - долю последовательных вычислений алгоритма: $f = 0.10$ (для $p=4$) и $f = 0.13$ (для $p=8$). Отсюда, применяя данный закон при $p \rightarrow \infty$, получим максимально возможное ускорение алгоритма, которое изменяется от 10раз (при использовании для расчётов $p=4$) до 7.7раз (используя $p=8$).

Заключение. В работе предложен эффективный алгоритм распределённого оделирования цифровых схем с неисправностями, использующий схему со статическим разбиением теста. Проведены машинные эксперименты, на основании которых оценена эффективность распараллеливания исходной задачи моделирования.

Литература

1. Niermann T.M., Cheng W.-T., Patel J.H. PROOFS: A Fast, Memory-Efficient Sequential Circuits Fault Simulator // IEEE Trans. CAD. – 1992.– P.198-207.
2. Kung C.P., Lin C.S. HyHope: A Fast Fault Simulator with Efficient Simulation of Hypertrophic Faults // Proc. of International Test Conference. - 1994. - P.714-718.
3. Иванов Д.Е., Скобцов Ю.А. Параллельное моделирование неисправностей для последовательных схем // Искусственный интеллект. – 1999. - №1. – С.44-50.
4. P.A. Duba, R.K. Roy, J.A. Abraham and W.A. Rogers, “Fault simulation in a distributed environment”, in Proceedings of the 25th ACM/IEEE Design Automation Conference, pp.686-691, June 1988.
5. T. Marcas, M. Royals and N. Kanopoulos, “On distributed fault simulation”, IEEE Computer, vol. 7, pp. 40-52, Jan. 1990.
6. S. Ghost, “NODIFS: a novel, distributed circuit partitioning based algorithm for fault simulation of combinational an sequential digital design on loosely coupled parallel processors”, tech. rep., LEMS, Division of Engineering, Brown University, Providence, RI, 1991.
7. Ладьженский Ю.В., Попов Ю.В. Программная система для исследования протоколов синхронизации при распределённом событийном логическом моделировании // Наукові праці Донецького національного технічного університету, Серія “Обчислювальна техніка та автоматизація”.- 2004.- Випуск №74.- С.201-209.
8. Brgles F., Bryan D., Kozminski K. Combinational profiles of sequential benchmark circuits // International symposium of circuits and systems, ISCAS-89. – 1989. – P.1929-1934.
9. Гергель, В.П., Стронгин, Р.Г. Основы параллельных вычислений для многопроцессорных вычислительных систем. - Н.Новгород, ННГУ, 2003, (2 изд.).

Работа поступила в редакцию 14.05.07.