

УДК 004.94

ВОЗМОЖНОСТИ ПАРАЛЛЕЛЬНОГО МОДЕЛИРОВАНИЯ ДИНАМИЧЕСКИХ СИСТЕМ СРЕДСТВАМИ MATLAB

Лысенко Д. В., Святный В. А.

Донецкий национальный технический университет

Анализируются недостатки современных средств моделирования, необходимость использования параллельного и распределенного моделирования. Рассматривается возможность моделирования динамических систем блочным методом с использованием пакета MATLAB. Оцениваются преимущества пакета MATLAB над другими средствами моделирования.

Введение

Моделирование как метод исследования объектов различной физической природы широко применяется в различных предметных областях, усложнение моделируемых объектов ведет к повышению требований к аппаратно-программным средствам реализации моделей. Возникший порог сложности, с которым не справляются последовательные ЭВМ с архитектурой фон Неймана, преодолевается применением вычислительных систем MIMD-архитектуры. Созданы программные средства в виде библиотек параллельных и распределенных вычислений (MPI), которые в соответствии с предписаниями стандартов используются с языками программирования C, C++, Fortran. Главным недостатком этих средств является низкий уровень дружелюбности к экспертам предметных областей, разрабатывающих математические модели; они должны осваивать соответствующий язык программирования. Вместе с тем, существующие языки моделирования, совместимые с пакетами MATLAB, Maple, Mathematica, Mathcad имеют более «высокоуровневый» сервис и облегчают как разработку, так и применение моделей, реализуемых на последовательных ЭВМ.

Рассмотрим вопросы организации параллельного моделирования в среде MATLAB.

1 Средства распределенного решения задач в MATLAB

В последних версиях пакета MATLAB [1] реализован инструментарий для параллельных вычислений (Parallel computing toolbox). Он позволяет выполнять три вида распараллеливания:

1. Вычисления на SIMD-архитектуре NVIDIA CUDA.
2. Многопоточные вычисления на одной многопроцессорной или многоядерной ЭВМ (так же можно получить прирост производительности с использованием технологии hyper-threading).
3. Распределенные вычисления на компьютерных кластерах.

Одной из особенностей среды MATLAB является ее ориентированность не на системного программиста, а на специалиста узкой предметной области. Синтаксис этого языка программирования, в отличие от языков общего назначения, приближен к математической записи. Изучение основ языка MATLAB для непрограммистов будет менее трудоемким, чем изучение, например, языка C++ с библиотеками для распределенных вычислений.

Параллельное программирование в среде MATLAB реализовано без необходимости использования специфических средств, характерных для языков общего назначения. Это привело к некоторому снижению гибкости и эффективности, но значительно упростило разработку программ для параллельных вычислений.

В данной статье будет более подробно рассмотрено распараллеливание на кластере и в рамках одного компьютера, способного эффективно работать в режиме многопоточности.

Распараллеливание вычислений происходит следующим образом. Запускается один процесс-клиент (client), который выполняет программу. В случае использования средств распараллеливания

процесс-клиент передает часть вычислительной работы процессам-рабочим (worker). Последние могут выполняться либо на одной машине с процессом-клиентом, либо на различных компьютерах в составе кластера. Кроме этого, в среде MATLAB реализована возможность распределенного хранения данных.

2 Блочнo-ориентированный подход к моделированию

Блочнo-ориентированное моделирование широко применялось на аналоговых вычислительных машинах. В роли функциональных блоков выступали аналоговые интеграторы, сумматоры, умножители. С развитием цифровой техники, вычислительная мощность цифровых ЭВМ стала достаточной для решения задач, которые ранее решались на аналоговых машинах. Работать на цифровых вычислительных машинах было проще, поэтому аналоговые ЭВМ были вытеснены.

Ранее цифровые ЭВМ развивались за счет улучшения технологического процесса производства интегральных микросхем, увеличения их быстродействия. В настоящее время повышение частоты работы процессоров прекратилось, повышение производительности достигается за счет использования более сложной архитектуры и распределенных вычислений. Разработка параллельных сред моделирования стала актуальной задачей [2].

Блочнo-ориентированное моделирование с использованием цифровых машин в какой-то мере является цифровым моделирование аналоговых моделей.

3 Пример блочно-ориентированного моделирования в среде MATLAB

Использование блочно-ориентированного подхода будет показано на примере генератора синусоиды (рис. 1) с использованием двух интеграторов и умножителя на константу (-1). На практике данная схема вряд-ли сможет где-либо использоваться, она была взята исключительно из-за простоты реализации и наглядности.

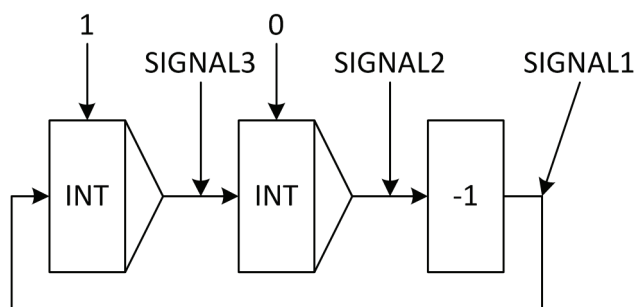


Рисунок 1. Используемая модель

Язык MATLAB поддерживает блочно-ориентированное программирование [3], эта технология была использована для моделирования. В данном случае используется три объекта – по одному для каждого функционального блока.

Листинг класса интегратора:

```
%BO integration object
classdef BO_INT < handle
    properties
        current_val
    end
    methods
        function INT = BO_INT(init_cond)
            INT.current_val=init_cond;
        end
        function step(bo_int,in,dt)
            bo_int.current_val=bo_int.current_val+in*dt;
        end
    end
end
```

```

        end
    end
end

```

Для этого класса определены две функции и одна переменная-параметр. Первая функция – конструктор, она вызывается при создании объекта и устанавливает начальные условия интегратора. Вторая функция осуществляет один шаг интегрирования, параметрами являются значение на входе интегратора и шаг интегрирования.

Подобным образом организован и класс множителя на константу:

```

%BO multiplication object
classdef BO_MUL < handle
    properties
        mul_val
        current_val
    end
    methods
        function MUL = BO_MUL(init_cond)
            MUL.mul_val=init_cond;
        end
        function step(bo_int,in)
bo_int.current_val=bo_int.mul_val*in;
        end
    end
end

```

Конструктор устанавливает константу, на которую происходит умножение, по функции step происходит умножение входного параметра на константу.

Для передачи информации между блоками используются сигналы и драйверы. Сигналы указывают значения входных данных для блоков, драйверы – выходных. В конце каждого шага моделирования сигналам присваиваются значения драйверов. Подобная организация сигналов используется в языке описания архитектуры VHDL.

Листинг основной программы моделирования:

```

%Блочно-ориентированный генератор синусоиды
clear
%частота
omega = 1;
delta_t = 0.001;
signal_val = [0 0 0];
signal_drv = [0 0 0];
int1 = BO_INT(0);
int2 = BO_INT(1);
mul1 = BO_MUL(-omega^2);
start_time=0;
step_number=7000;
X = zeros(1,step_number);
Y1 = zeros(1,step_number);
Y2 = zeros(1,step_number);
Y3 = zeros(1,step_number);
for i=1:step_number
    % 1st block
    int1.step(signal_val(1),delta_t);
    signal_drv(3) = int1.current_val;
    % 2nd block
    int2.step(signal_val(3),delta_t);

```

```

    signal_drv(2) = int2.current_val;
    % 3rd block
    mull.step(signal_val(2));
    signal_drv(1) = mull.current_val;
    % next step preparations
    signal_val = signal_drv;
    % logging results
    X(i) = start_time+i*delta_t;
    Y1(i) = signal_drv(1);
    Y2(i) = signal_drv(2);
    Y3(i) = signal_drv(3);
end
plot(X,Y1,X,Y2,X,Y3)

```

В начале происходит инициализация сигналов и драйверов (signal_val, signal_drv), создаются объекты для функциональных блоков (int1,int2,mull), задается шаг по времени и количество шагов (delta_t, step_number). Далее проходит непосредственно процесс моделирования, протоколирование и вывод результатов. Результаты моделирования представлены на рис.2.

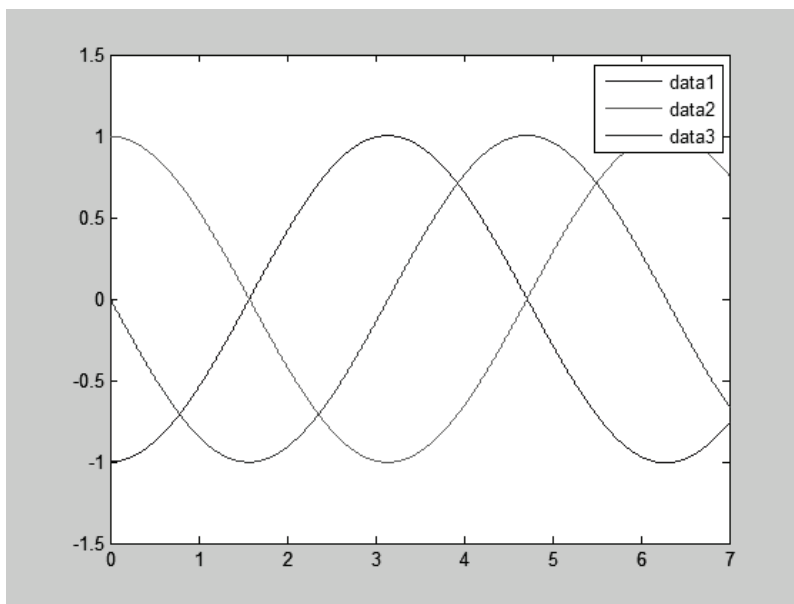


Рисунок 2. Результаты моделирования

В данной модели процедуры обработки для функциональных блоков (1st block, 2nd block, 3rd block) могут выполняться параллельно.

4 Перспективы для параллельной распределенной моделирующей среды

Распределенная параллельная моделирующая среда (РПМС) – системная организация аппаратных ресурсов, системного и моделирующего программного обеспечения, которая обеспечивает все этапы моделирования динамических систем.

Системное ПО в составе РПМС является сложной программной системой. Можно произвести ее декомпозицию на несколько подсистем:

1. Пользовательский интерфейс.
2. Подсистема анализа топологии.
3. Генератор уравнений.
4. Решатель уравнений.
5. Подсистема виртуальной параллельной имитационной модели.

6. Подсистема балансирования загрузки.
7. Решатель уравнений.
8. Подсистема визуализации.
9. IT – подсистема.
10. База данных.

В целом, среда MATLAB пригодна для реализации всех подсистем в составе системного программного обеспечения. С учетом особенностей синтаксиса языка MATLAB, реализация подсистем будет более простой по сравнению с реализацией на языках программирования общего назначения. В пункте 4 была показана возможность достаточно простой, лаконичной реализации блочно-ориентированной модели; наличие возможности организации распараллеливания вычислений.

Но одной из важнейших характеристик РПМС является ее эффективность, поэтому необходимы дальнейшие исследования эффективности работы инструментария для параллельных вычислений системы MATLAB.

Литература

- [1] Parallel Computing Toolbox – MATLAB. Электронный ресурс. Режим доступа: <http://www.mathworks.com/products/parallel-computing/>
- [2] Feldmann L.P., Svjatnyj V.A., Resch M., Zeitz M. FORSCHUNGSGEBIET: PARALLELE SIMULATIONSTECHNIK / Сайт национальной библиотеки им. В.И. Вернадского / Электронный ресурс. Режим доступа: <http://www.nbu.gov.ua/portal/natural/Npdntu/Pm/2008/08flpfps.pdf>
- [3] Andy H. Register. A guide to MATLAB object-oriented programming. – SciTech Publishing Inc. 2007