

ЭФФЕКТИВНОЕ ПРИМЕНЕНИЕ МНОГОЯДЕРНЫХ МИКРОПРОЦЕССОРОВ ДЛЯ РЕАЛИЗАЦИИ АЛГОРИТМОВ УПРАВЛЕНИЯ СТОХАСТИЧЕСКИМ ОБЪЕКТОМ¹

В.Н. Фетисов

Рассмотрена проблема применения многоядерных микропроцессоров для реализации алгоритма управления с прогнозирующей моделью марковским объектом. Показано, что для различных операционных систем реализовать параллельные вычисления можно довольно просто, если использовать соответствующие системные функции. Рассмотрены особенности применения предложенного подхода. В частности, указана область исходных данных, когда организация параллельных вычислений может оказаться неэффективной. Показана более высокая эффективность предлагаемого способа вычислений по сравнению с применением языка программирования OpenMP.

Ключевые слова: марковский объект, управление с прогнозирующей моделью, параллельные вычисления, многоядерный микропроцессор.

ВВЕДЕНИЕ

Традиционно словосочетание «параллельные вычисления и задачи управления» ассоциируется с проблемами, которые решаются с применением суперкомпьютеров или хотя бы с компьютерами, оснащенными десятками микропроцессоров. Конечно, методы и средства данного широкого научного направления не предназначены для управления объектами в реальном времени, но эффективны для использования на стадии проектирования. Данная работа нарушает установившуюся традицию. В статье предложено использовать параллельные вычисления непосредственно в алгоритме управления на стадии реализации. Сравнительно недорогие технические средства для этого уже имеются [1]. (Пока примитивные — на базе микропроцессоров всего с двумя ядрами). Но корпорация «Intel» обещает в скором времени разрабо-

тать микропроцессоры с ядрами более четырех для применения в промышленности.

Цель данной работы — представление простого и надежного метода использования параллельных вычислений при реализации алгоритмов управления стохастическим объектом.

Описание предлагаемого метода дано на примере алгоритма управления с прогнозирующей моделью для марковского объекта.

1. ПОСТАНОВКА ЗАДАЧИ

Главное достоинство алгоритмов управления с прогнозирующей моделью — гибкость перенастройки контроллера к изменениям математической модели объекта, а также простота реализации алгоритма при наличии ограничений на управление и выходную координату [2—6]. Недостаток этих алгоритмов в случае стохастического объекта заключается в громоздкости вычислений на каждом шаге управления, что может привести к невозможности практической реализации. Поэтому задача использования параллельных вычислений для алгоритма управления марковским объектом довольно актуальна [7, 8].

¹ Содержание статьи (в кратком изложении) было представлено на междунар. конф. «Параллельные вычисления и задачи управления» РАСО'2008, г. Москва, ИПУ РАН.



Сформулируем задачу управления. Пусть требуется минимизировать критерий качества

$$I(u) = \mathbf{M}\{Q(u_1, \dots, u_N, \dots)\} = \lim_{N \rightarrow \infty} \frac{1}{N} \mathbf{M} \left\{ \sum_{n=1}^N g(x_n, u_n) \right\},$$

где $\mathbf{M}\{\}$ — символ математического ожидания, x_n и u_n удовлетворяют уравнению

$$x_n = \phi(x_{n-1}, u_n) + \sigma \varepsilon_n, \quad x_0 = c,$$

где σ — положительное число, ε_n — независимые случайные величины с нулевым математическим ожиданием и единичной дисперсией, функции g и ϕ дифференцируемы по своим аргументам. Выходная координата объекта управления x_n наблюдается, u_1, \dots, u_N — управления.

В дальнейшем математическое ожидание критерия качества будем аппроксимировать с помощью метода Монте-Карло. Тогда алгоритм управления с прогнозирующей моделью определяется минимизацией на каждом шаге управления вспомогательного (скользящего) критерия качества вида

$$I_{x_{n-1}}(v) = \frac{1}{R} \sum_{r=1}^R \sum_{t=1}^T g(y_t^r, v_t) \quad (1)$$

по переменным v_t . Величины y_t и v_t связаны между собой с помощью уравнения, которое будем называть прогнозирующей моделью

$$y_t^r = \phi(y_{t-1}^r, v_t) + \sigma \varepsilon_t^r, \quad t_0^r = x_{n-1}.$$

Числа T и R в формуле (1) являются наименьшими, для которых выполняется неравенство

$$\mathbf{P} \left\{ \left| \frac{1}{R} \sum_{r=1}^R \left[\sum_{t=1}^T g(y_t^r, v_t) - \lim_{N \rightarrow \infty} \sum_{t=1}^T g(y_t^r, v_t) \right] \right| > \alpha \right\} \leq \delta,$$

где величины α и δ заданы.

Обозначим v_1^*, \dots, v_N^* управление, дающее минимум критерия (1). В качестве управляющего воздействия на n -м шаге положим $u_n^* = v_1^*(x_{n-1})$. На шаге $n + 1$ после минимизации вспомогательного критерия положим $u_{n+1}^* = v_1^*(x_n)$ и на следующих шагах будем действовать по аналогичному правилу. Последовательность $\{u_n^*\}$ и определяет алгоритм управления с прогнозирующей моделью.

Процедура минимизации семейства функционалов $I_{x_{n-1}}(v)$, $n = 1, 2, \dots$, должна осуществляться в реальном времени непосредственно в процессе управления объектом. В отличие от синтеза систе-

мы управления, когда множество управлений принадлежит функциональному пространству, в системах управления с прогнозирующей моделью множество управлений определено в конечномерном линейном пространстве, что существенно упрощает решение задачи управления.

Главная проблема минимизации функционала (1) связана с вычислительным аспектом задачи, поскольку теперь все вычисления должны уложиться во временной интервал допустимого по требованиям точности шага управления.

Решение задачи управления (1) обычно невозможно получить в аналитическом виде. Поэтому применяются рекуррентные методы оптимизации. Без нарушения общности далее будет рассмотрен метод градиента [8].

Для упрощения математических выражений и без нарушения общности рассмотрим оптимизацию критерия (1) на одном временном интервале. Тогда получим

$$I(v) = \frac{1}{R} \sum_{r=1}^R \sum_{t=1}^T g(y_t^r, v_t), \quad (2)$$

$$y_t^r = \phi(y_{t-1}^r, v_t) + \sigma \varepsilon_t^r, \quad t_0^r = c.$$

Далее индекс n будет обозначать номер шага рекуррентной оптимизации. Имеем

$$V_n = V_{n-1} - \mu \text{grad}_n, \quad V_0 = 0,$$

где μ — положительное число, V_n — вектор управлений размерности T , а компоненты вектора grad_n^k , $k = 1, \dots, T$, определяются формулой

$$\text{grad}_n^k \approx [I(v_{1n}^1, \dots, v_n^k + \Delta, \dots, v_n^T) - I(v_n^1, \dots, v_n^k, \dots, v_n^T)]/\Delta, \quad (3)$$

где Δ — положительное число.

Известно [8], что имеет место равенство

$$\lim_{\Delta \rightarrow 0, R \rightarrow \infty, n \rightarrow \infty} V_n = V.$$

Задача рекуррентной минимизации критерия качества (2) довольно громоздка в вычислительном аспекте.

Идеи использования решения громоздких в вычислительном аспекте задач на компьютерах с несколькими процессорами появились давно [9]. В работе [10] была предложена архитектура микропроцессора с несколькими модулями операций с плавающей запятой (FPU), позволяющая осуществлять параллельные вычисления для широкого класса задач управления. Корпорация «Intel» для реализации параллельных вычислений выбрала другой (более общий) путь и в 2006 г. объявила

о создании микропроцессора с двумя «ядрами» (а несколько позже с четырьмя), в составе которых оказались почти все составляющие обычного микропроцессора. В настоящее время корпорация «Intel» приступила к разработке микропроцессоров с числом ядер исчисляемым десятками. Многоядерная архитектура совместно с некоторыми изменениями системных функций ряда операционных систем привела к возможности использовать стандартные языки программирования для реализации параллельных вычислений. Корректное написание программы позволяет почти в два раза для двух ядер и почти в четыре раза для четырех ядер увеличить скорость вычислений.

Далее будет предложен простой способ программной реализации алгоритма управления с марковской прогнозирующей моделью для микропроцессора с несколькими ядрами. Рассмотрен пример решения такой задачи с помощью четырехъядерного процессора и показана степень эффективности параллельного программирования в зависимости от размерности вектора градиента.

2. РЕАЛИЗАЦИЯ АЛГОРИТМА УПРАВЛЕНИЯ ДЛЯ МНОГояДЕРНЫХ МИКРОПРОЦЕССОРОВ

Вычисление вектора градиента в алгоритме управления занимает более 99 % от всего расчетного времени. Используя закон Амдала [9] для расчета коэффициента увеличения производительности β :

$$\beta \leq \frac{1}{\omega + (1 - \omega)n^{-1}},$$

где ω — доля программного кода без параллельных расчетов и n — число ядер, получим для четырехъядерного процессора предельное увеличение производительности в 3,9 раза. Как будет показано далее, увеличение производительности часто может быть более низким.

Перепишем вектор (3) в детализированном виде (назовем такое представление факторизацией вектора градиента)

$$\text{grad}_n^k \approx \frac{1}{R} \sum_{r=1}^R \left[\sum_{t=1}^T g(y_{in}^r, v_n^k + \Delta, \dots, v_n^T) - \sum_{t=1}^T g(y_{in}^r, v_{in}^t) \right] / \Delta, \quad (4)$$

где переменные y_{in}^r и v_n^k связаны уравнением

$$y_{in}^r = \phi(y_{(t-1)n}^r, v_{in}^k + \xi) + \sigma \varepsilon_t^r, \quad t_0^r = c.$$

Здесь вектор ξ размерности T имеет вид

$$\xi_t = \begin{cases} \Delta, & t = k, \\ 0, & t \neq k. \end{cases}$$

Как видно из формулы (4), слагаемые суммы по индексу r (выражения в квадратных скобках) не связаны между собой и могут быть рассчитаны параллельно при наличии многоядерного микропроцессора (или нескольких обычных микропроцессоров).

Пусть M ($M \leq R$) обозначает число ядер микропроцессора. Разобьем сумму по r на части. Всего таких частей будет R/M , если R делится на M без остатка, и $[R/(M-1)] = 1$ в противном случае. Таким образом, распределив вычисления полученных частичных сумм по ядрам, теоретически можно в M раз увеличить расчетную скорость.

Существует ряд языков параллельного программирования высокого уровня [11–14], которые разработаны для различных операционных систем — ОС (Windows, LUNIX и др.) Один из таких языков, OpenMP, наиболее распространен. (Официально он называется «стандартом программирования» и первоначально предназначался для программирования на суперкомпьютерах.)

На рис. 1 [14] показан график увеличения производительности двухъядерного процессора благодаря использованию параллельных вычислительных потоков с применением языка OpenMP при вычислении суммы вида

$$x = \sum_{i=1}^R (y_{i-1} + y_{i+1})/2,$$

где последовательность $\{y_i\}$ независима. По оси абсцисс отложена величина R , по оси ординат — относительное увеличение производительности.

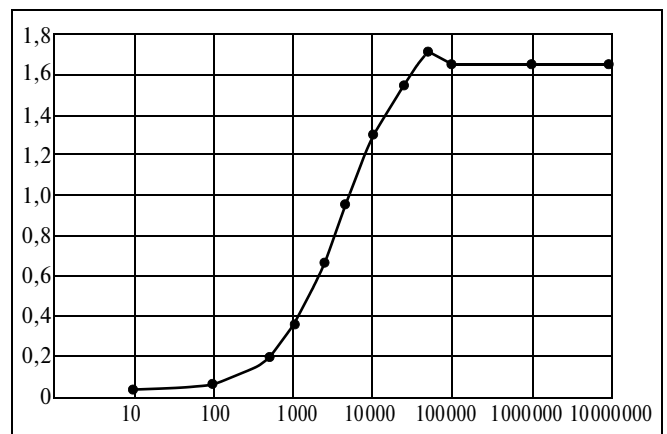


Рис. 1. Улучшение производительности с использованием языка OpenMP



Теперь покажем, как можно организовать параллельные вычисления, используя язык программирования Си и системные функции современных операционных систем (функции API в случае ОС «Windows»).

Далее представлен фрагмент программы реализации параллельных потоков, который был использован при решении задачи оптимизации с применением четырехъядерного микропроцессора (операционные системы x64 «Microsoft»):

```
HANDLE hThread[MAX_THREADS];
LPVOID npTTYInfo = NULL;
DWORD dwThreadID[MAX_THREADS];
//MAX_THREADS обозначает число ядер//
//Инициализируем параллельные потоки://
hThread[0]=CreateThread(
    (LPSECURITY_ATTRIBUTES) NULL,
    0,
    (LPTHREAD_START_ROUTINE)M0,
    (LPVOID) npTTYInfo,
    0, &dwThreadID[0]);
hThread[0]=CreateThread(
    (LPSECURITY_ATTRIBUTES) NULL,
    0,
    (LPTHREAD_START_ROUTINE)MAX,
    (LPVOID) npTTYInfo,
    0, &dwThreadID[0]);
//Ждем окончания вычисления в потоках://
WaitForMultipleObjects(MAX_THREADS-1,
    hThread, TRUE, INFINITE);
//Далее реализация вычислений в
//параллельных потоках//
DWORD WINAPI M0(LPVOID npTTYInfo)
{
    //Вычисление градиента для условия 0//
    return 0;
}
DWORD WINAPI MAX(LPVOID npTTYInfo)
{
    //Вычисление градиента для условия MAX//
    return MAX_THREADS-1;
}
```

В случае операционной системы UNIX вместо системной функции WaitForMultipleObjects используется следующий фрагмент программы:

```
for(k = 0; k < MAX_THREADS; k++)
pthread_join(dwThreadID[k], NULL).
```

Таким образом, в случае применения предложенного метода, для организации параллельных вычислений независимо от применяемой ОС используются всего две специальные системные функции.

При программировании нескольких параллельных вычислительных потоков желательно соблюдать следующие правила:

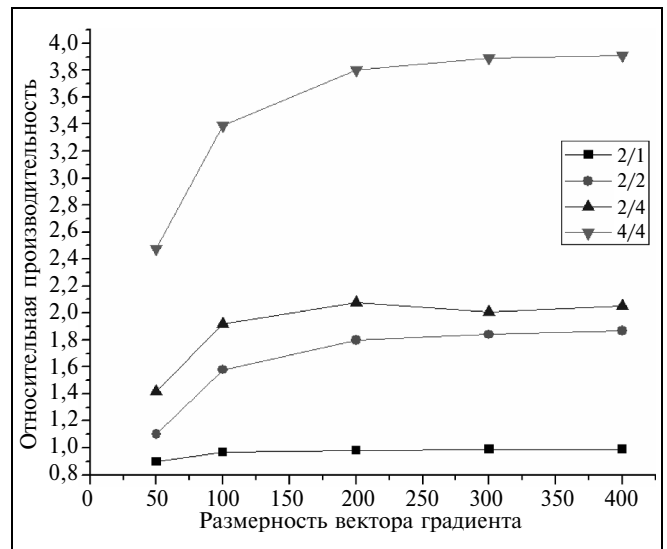


Рис. 2. Результаты моделирования для исследования относительной производительности

- глобальные переменные для различных потоков должны иметь уникальные идентификаторы;
- по возможности не пользоваться стандартными математическими библиотеками;
- прототипы математических функций определять со спецификатором «inline».

Рассмотренный в данной работе подход к организации параллельных вычислений был реализован на примере задачи управления нелинейным марковским объектом второго порядка вида

$$y_t = a_1 y_{t-1} + a_{21} y_{t-2} + a_{22} y_{t-2}^2 + u_t + \sigma \varepsilon_t,$$

$$y_0 = c_0, \quad y_1 = c_1$$

с квадратичным критерием качества.

Здесь y_t — наблюдаемая выходная координата, t — дискретное время, u_t — управление, ε_t — последовательность независимых случайных величин, a_1, a_{21}, a_{22} и σ — константы.

На рис. 2 представлен график, показывающий увеличение относительной производительности вычисления вектора градиента в зависимости от его размерности при использовании предложенного подхода реализации параллельных вычислений.

Здесь отношение чисел, указанное для каждой кривой, означает отношение числа параллельных вычислительных потоков к числу ядер. Для малых значений размерности вектора градиента низкая производительность параллельных вычислений объясняется громоздкостью выполнения системных функций, организующих параллельные потоки.

Добавим, что имеет смысл повышать производительность именно в ситуации, когда произво-

дятся громоздкие вычисления. При малом объеме вычислений проблема повышения производительности вычислителя обычно не столь важна. Из сравнения рис. 1 и 2 можно сделать вывод, что метод системных функций более эффективен, чем использование языка OpenMP.

Для ОС Vista, Red Hat Enterprise 5 и Solaris 10 результаты моделирования практически совпали. Все эксперименты проводились на 64-разрядном микропроцессоре «Intel Core 2 Quad» с четырьмя ядрами.

ЗАКЛЮЧЕНИЕ

Решение задачи управления с прогнозирующей моделью марковским объектом весьма громоздко в вычислительном аспекте. Показано, что алгоритм оптимизации на каждом шаге управления может быть реализован с помощью параллельных вычислительных потоков. Параллельные вычислительные потоки можно реализовать различными способами. Один из альтернативных способов организации параллельных вычислений предложен в работе [10].

Представлен способ реализации алгоритма оптимизации на компьютере с многоядерным микропроцессором. Рассмотрен простой и наиболее естественный способ организации параллельных вычислений для различных операционных систем.

Анализируется, в каких случаях организация параллельных вычислений приводит к эффективному решению для многоядерного микропроцессора, и в каких случаях параллельные вычисления перестают быть эффективными. Показана более высокая эффективность предлагаемого способа вычислений по сравнению с применением языка программирования OpenMP.

ЛИТЕРАТУРА

1. ARK-3399 — High-performance Compact Embedded Computer with Intel® Core™ Duo/Core™ 2 Duo. URL: <http://www.advantech.com/products/> (дата обращения: 29.10.2008).
2. Буков В.Н. Адаптивные прогнозирующие системы управления полетом. — М.: Наука, 1987.
3. Filatov N.M., Unbehauen H. Adaptive predictive control policy for nonlinear stochastic systems // IEEE Trans. on automatic control. — 1995. — Vol. 40, N 11. — P. 1943—1949.
4. Camacho E.F., Bordons C. Model Predictive Control. — London: Springer-Verlag, 1999.
5. Constrained model predictive control: Stability and optimality (Survey Paper) / D.Q. Mayne, J.B. Rawlings, C.V. Rao, O.M. Scokaert // Automatica. — 2000. — Vol. 36. — P. 789—814.
6. Фетисов В.Н. Точность стохастической системы управления с детерминированной моделью в качестве прогнозатора // Автоматика и телемеханика. — 2002. — № 12. — С. 49—58.
7. Параллельные вычисления и задачи управления (аналитический обзор) / А.Л. Бунич, К.С. Гинсберг, А.В. Добровидов и др. // Автоматика и телемеханика. — 2002. — № 12. — С. 3—23.
8. Поляк Б.Т. Введение в оптимизацию. — М.: Наука, 1983.
9. Amdahl G. Validity of the single processor approach to achieving large scale computing capabilities, AFIPS Conference Proceedings. — 1967. — Vol. 30. — P. 483.
10. Фетисов В.Н. Перспектива разработки и использования микропроцессоров с параллельной архитектурой устройства обработки чисел с плавающей запятой // Тр. междунар. конф. «Параллельные вычисления и задачи управления» РАСО' 2004. Москва 2 — 4 октября 2004 г. / ИПУ РАН. — М., 2004. — С. 182—188.
11. Левин М.П. Параллельное программирование с использованием OpenMP. — М.: Бином, 2008.
12. Рахул В. Патил и Бобби Джордж. Средства и приемы для выявления проблем параллельного выполнения // MSDN Magazine. — 2008. — N 8.
13. Intel Threading Building Blocks (Tutorial), Intel Corporation, 2007. URL: <http://www.intel.com>. (дата обращения: 16.01.2008).
14. Kang Su Gatlin, Pete Isensee. Reap the Benefits of Multithreading without All the Work // MSDN Magazine. — 2005. — N 10.

Статья представлена к публикации членом редколлегии В.Д. Малюгиным

Фетисов Вячеслав Николаевич — д-р техн наук, профессор, Московский технический университет связи и информатики, ✉ vfetisov@stream.ru.

Новая книга

Дартау Л.А., Мизерницкий Ю.Л., Стефанюк А.Р. Здоровье человека и качество жизни: проблемы и особенности управления. — М.: СИНТЕГ, 2009. — 400 с.

Проблемы медико-социального благополучия населения, здоровья и качества жизни рассмотрены с позиций классической теории автоматического управления. Обоснована ключевая роль индивида в сохранении здоровья. Показано, что непрерывное обучение здоровью и эпизодическое лечение являются основой «воздействия на объект» в цепи прямой связи, а перманентное выявление факторов риска заболеваний — основой для коррекции как управляющих воздействий, так и образа жизни индивида в этико-правовых рамках нового общественного договора о социальном партнерстве. Сделан вывод, что ответственность за конечный результат (индивидуальное и популяционное здоровье и качество жизни) должна быть поделена между всеми субъектами управления и основана на межсекторальном сотрудничестве, которое обеспечивается администрацией в местах проживания, работы или учебы. На основе анализа документов Всемирной организации здравоохранения, касающихся современного состояния здоровья населения развитых стран, предложен возможный вариант реализации ее рекомендаций в России.

Руководителям структур государственного, территориального и муниципального управления, организаторам здравоохранения и образования, а также широкому кругу читателей, заинтересованных в информации о современных возможностях и ограничениях в управлении здоровьем и качеством жизни.