УДК 004.451.45

Ю.А. Иванов, канд. техн. наук, доцент,
А.Ю. Иванов, старший преподаватель
Донецкий национальный технический университет
ai@cs.dgtu.donetsk.ua

Динамическая модель подвижного состава на многоядерной вычислительной системе

Рассмотрена задача анализа динамики основных частей подвижного состава в вертикальной продольной плоскости. Выполнена дискретизация модели колебаний сложной динамической системы. Обоснована, разработана и исследована программная модель колебаний элементов подвижного состава на многоядерной системе.

Ключевые слова: динамическая система, многоядерная система, SMP система, программная модель, подвижной состав.

Введение

Распространение многоядерных процессоров и внедрение их в практику вычислений предоставляет разработчикам новые возможности по расширению и повышению сложности компьютерных моделей динамических систем. Необходимые средства многопоточного программирования присутствуют в распространенных операционных системах, однако объем специальных знаний, необходимых разработчикам моделей на параллельных компьютерных платформах, не позволяет им сосредоточиться на вопросах собственно исследования моделей. Поэтому вопросы создания и эффективной реализации многопоточных моделей для многоядерных систем являются актуальной задачей.

Современные многоядерные архитектуры предоставляют разработчикам параллельное программное обеспечение на аппаратных платформах разные средства: API Microsoft для Win32, API POSIX-потоки и API OpenMP. Программное обеспечение Intel предлагает множество различных ресурсов, один из которых – база знаний с техническими описаниями и статьями (http://www.intel.com/multi-core/) Intel, которая описывает предлагаемое правильное понимание оптимальных поточных технологий [1].

Операционные системы семейства Microsoft Windows обеспечивают работу вычислительной системы с симметричной мультипроцессорной архитектурой (SMP), при которой все процессоры имеют прямой и равноправный доступ к любой точке общей памяти. В статья приводятся результаты исследования возможностей, предоставляемых многоядерными процессорами, поддерживающими низкоуровневые многопоточные технологии (поточный прикладной программный интерфейс) операционных систем Windows для создания модели динамической системы. Рассматривается необходимость анализа качества параллельной программной реализации модели средствами инструментов разработки Microsoft и Intel, для дальнейшего практического применения предложенного способа разработки в конкретных натурных моделях. Составлены рекомендации по учету аппаратных особенностей многоядерных систем при разработке таких моделей.

1. Модель динамической системы

Эксплуатация скоростного подвижного состава предъявляет повышенные требования к конструкции и содержанию локомотивов и железнодорожного пути. В этой связи требования безопасности, динамическим качествам локомотивов и подвижному составу на сегодняшний день должны быть жёсткими [2,3].

Уравнения движения рассматриваемой системы вертикальных колебаний рельсовых экипажей с двухъярусным подвешиванием: магистральные локомотивы (электровозы и тепловозы) и пассажирские вагоны описывается следующими дифференциальными уравнениями [3].

Уравнения колебаний кузова:

$$m_k \cdot z_k'' + b_k(2z_k' - z_1' - z_2') + c_k(2z_k - z_1 - z_2) = 0. \quad (1)$$

$$J_k \cdot \varphi_k'' + a_k \cdot b_k(2a_k \cdot \varphi_k' - z_1' + z_2') + a_k \cdot c_k(2a_k \cdot \varphi_k - z_1 + z_2) = 0 \quad (2)$$

Уравнения колебаний первой тележки

$$m_r \cdot z_1'' - b_k(z_k' - z_1' + a_k \cdot \varphi_k') - c_k(z_k - z_1 + a_k \cdot \varphi_k) + 2b_r \cdot z_1' + 2c_r \cdot z_1 = b_r(\eta_1' + \eta_2') + c_r(\eta_1 + \eta_2). \quad (3)$$

Уравнения колебаний второй тележки

$$m_r \cdot z_2'' - b_k(z_k' - z_2' + a_k \cdot \varphi_k') - c_k(z_k - z_2 + a_k \cdot \varphi_k) + 2b_r \cdot z_2' + 2c_r \cdot z_2 = b_r(\eta_3' + \eta_4') + c_r(\eta_3 + \eta_4), \quad (4)$$

при следующих обозначениях и их значениях: m_k = 57 т – масса кузова, J_k = 70 – момент инерции кузова, m_T = 9 т – масса тележки, a_k = 3,725 м – половина базы кузова, c_T = 3040 кН/м, b_T = 30 кНс/м – жесткость и демпфирование в первом ярусе подвешивания (тележ-

ка), c_k = 2660 кН/м, b_k = 100 кН.с/м – жесткость и демпфирование во втором ярусе подвешивания (кузов), z_i, z_i', z_i'', φ_k, φ_k', φ_k'' – обобщенные координаты и их производные по времени, $\eta_j(t)$ – возмущение со стороны пути под j-й колесной парой. В качестве возмущения со стороны пути использована усреднённая двугорбая геометрическая неровность Н. Н. Кудрявцева, описываемая уравнением

$$\eta(t) = A1\sin(wt) + A2\sin(2wt). \qquad (5)$$

Модель неровности представляет собой сумму полуволны синусоиды частотой *(wt)* и трех полуволн синусоиды частотой *(3wt)*, уложенные на длине рельсового звена.

Исследуемое внешнее возмущение для системы с транспортным запаздыванием $\eta_i=(t-\tau_i)$, определяется геометрическими размерами и скоростью движения V

$$\begin{aligned} w &= 2\pi V / L \\ n_1(t) &= n_0(\sin wt) \\ n_2(t) &= n_1(t)(t - 2a_1 / V) \\ n_3(t) &= n_1(t)(t - 2a_2 / V) \\ n_4(t) &= n_1(t)(t - 2(a_1 + a_2)/V) \\ n_{11}(t) &= n_0 w(\cos wt) \\ n_{22}(t) &= n_1(t)(t - 2a_1 / V) \\ n_{33}(t) &= n_{11}(t)(t - 2a_2 / V) \\ n_{44}(t) &= n_{11}(t)(t - 2(a_1 + a_2)/V). \end{aligned} \qquad (6)$$

2. Исследование модели динамической системы средствами MATLAB

Дифференциальные уравнения динамической системы (1-6) сведены к системе уравнений первого порядка методом понижения порядка производной. Уравнения динамической системы, приведенные к виду, подходящему для интегрирования в *MATLAB* имеют вид, представленный на рис.1.

Для получения результатов решения (рис.2) использован адаптивный метод – функция ode45, при этом дополнительно к указанным, использованы следующие исходные данные: *n0=0,005; L=25; a_1=1,5; a_2=3,725.*

```
D(1)=x(2);     % 1 trolley
D(2)=1/m1*(b1*(n11+n22-2*x(2))+c1*(n1+n2-
2*x(1))+b2*(x(6)-x(2)+a2*x(8))+c2*(x(5)-
x(1)+a2*x(7)));
D(3)=x(4);     % 2 trolley
D(4)=1/m1*(b1*(n33+n44-2*x(4))+c1*(n3+n4-
2*x(3))+b2*(x(6)-x(4)-a2*x(8))+c2*(x(5)-
x(3)-a2*x(7)));
D(5)=x(6);     % carcas
D(6)=1/m2*(b2*(x(2)+x(4)-
2*x(6))+c2*(x(1)+x(3)-2*x(5)));
D(7)=x(8);
D(8)=a2/j2*(b2*(x(2)-x(4)-
2*x(8)*a2)+c2*(x(1)-x(3)-2*x(7)*a2));
```

Рисунок 1 – Система (1-5), приведенная для решения в среде *MATLAB*

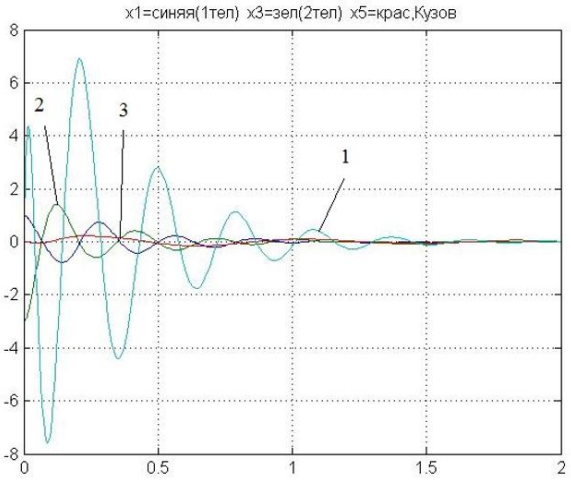

Рисунок 2 – Изменение скоростей перемещений z_1', z_3', z_5' системы в среде моделирования *MATLAB*

3. Программная модель системы

Для получения эффективной параллельной модели необходимо учесть ограничения: по заданиям, по данным, по информационным потокам. Управление одновременными действиями и их возможным взаимовлиянием приводит к проблемам четырех типов: синхронизации, взаимодействия, балансировки нагрузки и масштабируемости. Декомпозиция модели и управление одновременными действиями может быть проведена несколькими известными способами, имеющими свои особенности.

На первом этапе исследования был выбран способ распределения уравнений подобный аналоговому программированию, т.е. каждое уравнение модели было распределено на отдельное ядро системы. Вычислительная схема соответствует системе уравнений модели 1-6, сведенной к дифференциальным уравнениям первого порядка.

В основу создания модели динамической системы положен параллелизм на уровне потоков (Thread level Parallelism, TLP). С помощью функции CreateThread в процессе создаются 4 потока для интегрирования уравнений системы соответственно:1 и 5, 2 и 6, 3, 4. С помощью функции SetThreadAffinityMask устанавливаются маски родственности каждого из четырех ядер созданных потоков. Каждый из потоков имеет значение параметра CREATE_SUSPENDED и в дальнейшем они запускаются вызовом функции ResumeThread**.** Для минимизации влияния накладных расходов использованы функции SetThreadPriority со значением приоритета THREAD_PRIORITY_ABOVE_NORMAL. Завершение моделирования обеспечивается функцией WaitForMultipleObjects по окончанию интегриро-

вания всех потоков на соответствующих ядрах. Исполнение потока завершается, когда функция потока возвращает управление. Функции CreateThread создают поток в контексте процесса моделирования, из которого они вызваны. Поскольку потоки модели расположены внутри одного процесса, они совместно используют его адресное пространство (один контекст памяти) и имеют доступ к глобальным переменным приложения [5,6]. Для локальных переменных потоков система обеспечивает разные физические адреса для каждого потока в диапазоне виртуальных адресов секции .tls. Принятое распределение уравнений модели между ядрами привело к тому, что к глобальным переменным приложения отнесены переменные обмена между блоками, код вычисления которых образует критические секции.

Основной трудностью многопоточного программирования является реализация критических секций таким образом, чтобы потоки не использовали секции одновременно. При реализации механизмов синхронизации рекомендуется минимизировать число блокировок чтения-записи, что обеспечивает одновременное чтение для нескольких потоков, но запись разрешают только одному потоку. В случае многоядерной многопоточной программной модели динамической системы становится не столь важным размер критической секции и сложности синхронизации переносятся на информационное взаимодействие потоков через кэши L1, L2, L3. Обращения, обслуживаемые из кэша, не занимают полосу пропускания шины памяти, но создают проблему когерентности кэшей [1]. Поэтому в модели использован способ выравнивания (padding) данных для переменных обмена между ядрами. При этом были учтены системные измерения времени на каждом ядре программной секцией помощью API-функций QueryPerformance Frequency, QueryPerformanceCounter.

Многопоточный код требует проверки на корректность исполнения. Ошибки работы с памятью и многопоточности редко проявляются и трудно отлаживаемы. При разработке модели был использован анализатор корректности Intel® Inspector XE (рис. 3) из набора Intel® Parallel Studio XE 2013.

Данный инструмент динамического анализа памяти и многопоточности находит уязвимые места в коде, утечки и ошибки доступа к памяти, приводящие к «зависаниям» и «падениям», конфликты потоков и блокировки в многопоточной программе (тесты многопоточные), поддерживает различные реализации (в том числе простые потоки) моделей многопоточности, работает со стандартными сборками и библиотеками и интегрируется в Microsoft Visual Studio [7].

Тестирование с помощью данного инструмента позволило проверить наличие ошибок доступа, утечки памяти и конфликтов потоков в многоядерной многопоточной программной модели динамической системы. Таким образом, использованная утилита позволяет ускорить процесс разработки и отладки программных реализаций динамических моделей, ориентированных на исполнение на многоядерных вычислительных системах.

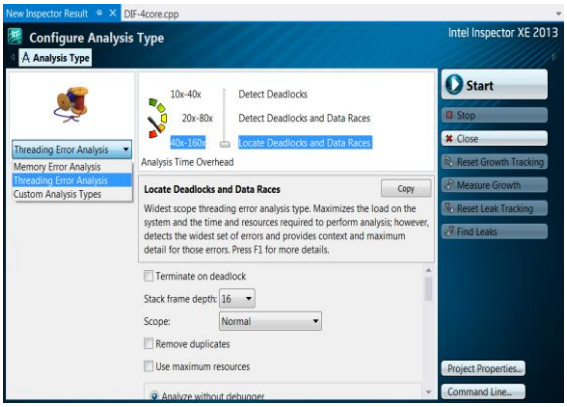

Рисунок 3 – Окно Intel Inspector XE при отладке программы модели

Результаты численных экспериментов (рис.4) для многоядерной многопоточной программной модели проведённых для скоростей 20 – 150 км/ч совпали с результатами исследования средствами MATLAB и упрощенной модели, реализованной на кластере [4].

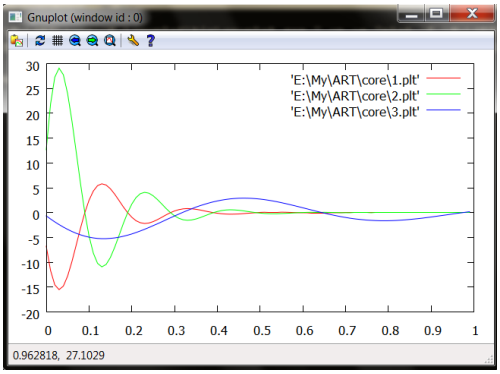

Рисунок 4 – График зависимости перемещений z_2, z_4, z_6 во времени в многопоточной модели

Заключение

Появление новых режимов работы динамических объектов, развитие технологий и техники выдвигает дополнительные требования к созданию компьютерных моделей. В работе проанализированы проблемы создания эффективных параллельных моделей для исполнения на многоядерных процессорах. Исследованы стандартные возможности операционной системы Windows по управлению выполнением программ на SMP системах с учетом ограничений по синхронизации и

взаимодействию между потоками. Установлено, что использование инструмента Intel Inspector XE (на предмет наличия в программной реализации модели «узких мест»: блокировок, гонок данных) позволяет сократить время разработки параллельной модели.

Разработанная программная модель может рассматриваться как архитектурное решение по организации системного обеспечения модели предназначенного для задач анализа переходных (нормальных и аварийных) режимов эксплуатации и проектирования подвижного состава.

Список литературы

1. Эхтер Ш. Многоядерное программирование / Ш. Эхтер, Д. Робертс. – СПб.: Питер, 2013. – 316 с.
2. Токмурзина Н. А. Особенности экипажной части и динамические характеристики электровоза KZ4A. / Н. А. Токмурзина, Пя Д. Р. // Материалы всероссийской научно-технической конференции с международным участием "Технологическое обеспечение ремонта и повышение динамических качеств железнодорожного подвижного состава". – Омск: Омский государственный университет путей сообщения, 2011. – 337 с.
3. Матва А. М. Решение задач динамики железнодорожных экипажей в пакете MATHCAD : учеб. пособие / В.Г. Рубан, А.М. Матва. – Ростов-на-Дону: Ростовский государственный университет путей сообщения, 2009. – 99 с.
4. Середа О. Э. Кластерная модель динамической системы / О.Э. Середа, А.Ю. Иванов // Інформаційні управляючі системи та комп'ютерний моніторинг (ІУС КМ - 2013) : IV Всеукраїнська науково-технічна конференція студентів, аспірантів та молодих вчених, 24-25 квітня 2013 р. – Донецьк: ДонНТУ. – 2013. – Т.1. – С. 247–251.
5. Русинович М. Внутреннее устройство Microsoft Windows : 6-е изд. / М. Русинович, Д. Соломон. – СПб.: Питер, 2013. – 800 с.
6. Харт М. Д. Системное программирование в среде Windows. Третье издание / М. Д. Харт. – М.: Вильямс, 2005. – 592 с.
7. Описание программного продукта Intel® Parallel Studio XE 2013 [Электронный ресурс]. – 2013. – Режим доступа к порталу: http://software.intel.com/en-us/intel-parallel-studio-xe/.

Надійшла до редакції 11.03.2014

Ю.О. ІВАНОВ[1], О.Ю. ІВАНОВ[1]
[1] Донецький національний технічний університет

РОЗРОБКА БАГАТОЯДЕРНОЇ МОДЕЛІ ДИНАМІКИ РУХОМОГО СКЛАДУ

Розглянута задача аналізу динаміки основних частин рухомого складу у вертикальній подовжній площині. Обгрунтована та виконана дискретизація моделі коливань складної динамічної системи для багатоядерної платформи. Розроблена програмна модель руху коливань елементів рухомого складу для багатоядерної системи.

Ключові слова: динамічна система, багатоядерна система, SMP система, програмна модель, рухомий склад.

Y.O. IVANOV[1], O.Y. IVANOV[1]
[1] Donetsk National Technical University

DEVELOPMENT OF A MULTI-CORE DYNAMIC MODEL OF ROLLING STOCK

In the article we consider the problem of analyzing dynamics of main parts of rolling stock in the vertical longitudinal plane. We performed sampling of fluctuation model of a complex dynamic systems for multi-core platform. A software model of motion of rolling elements vibrations for multi-core systems was developed. In this paper we analyze the problem of creating effective models for parallel execution on multicore processors. Necessary instruments for multithread programming are presented in common operating systems, however, the amount of expertise required for simulation on parallel computing platforms, does not allow itself to focus on the issues of research models. Therefore, the establishment and effective implementation of multi-core multithreaded models are urgent task. For efficient parallel model it is important to take into account limitations: each task, data, information flows. In this paper at this stage of study we chose a method to distribute equations similar to analog programming, so each equation of the model was distributed on a separate core of the system. The article investigates the standard features of the Windows operating system to manage program execution on SMP systems with the constraints of the interaction and synchronization between threads. Effectiveness of the model was evaluated using Intel Inspector XE tools for software implementation to determine presence of "bottlenecks": locks, data races. Results of numerical experiments for multi-core multi-threaded programming model conducted for speeds of 20 - 150 km/h were compared with the results of research performed using a simplified model in MATLAB and implemented on the cluster.

Keywords: dynamic model, multi-core system, SMP system, model, rolling stock.