

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ДИНАМІКИ ТЕПЛОВИХ ПРОЦЕСІВ В АПАРАТНИХ КОМПОНЕНТАХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМ

Вінницький національний технічний університет

Анотація

У роботі досліджено математичні засади моделювання теплових процесів у сучасних мікропроцесорах з використанням диференціальних рівнянь першого порядку. Проведено порівняльний аналіз ефективності повітряних та рідинних систем охолодження на основі закону охолодження Ньютона. Запропоновано підходи до програмної оптимізації керування вентиляторами для запобігання температурному тротлінгу. На основі отриманої математичної моделі розроблено програмний алгоритм мовою C++ для предиктивної діагностики (Predictive Maintenance) апаратного забезпечення, що дозволяє виявляти деградацію систем охолодження до настання критичного перегріву [3].

Ключові слова: диференціальні рівняння, закон Ньютона, системне програмування, C++, предиктивна діагностика, тротлінг, апаратні компоненти.

Abstract

The paper investigates the mathematical foundations of modeling thermal processes in modern microprocessors using first-order differential equations. A comparative analysis of the efficiency of air and liquid cooling systems is carried out based on Newton's law of cooling. Approaches to software optimization of fan control to prevent thermal throttling are proposed. Based on the obtained mathematical model, a C++ software algorithm for predictive maintenance of hardware is developed, which allows detecting the degradation of cooling systems before critical overheating occurs.

Keywords: differential equations, Newton's law, systems programming, C++, predictive maintenance, throttling, hardware components.

Вступ

Стрімке зростання продуктивності сучасних обчислювальних систем супроводжується значним збільшенням тепловиділення центральних (CPU) та графічних (GPU) процесорів. При досягненні критичних температур (зазвичай 95°C для AMD Ryzen та 100–105°C для сучасних поколінь Intel Core) активується механізм теплового тротлінгу – примусового зниження тактової частоти для захисту кристала від деградації. Для розробки ефективних алгоритмів автоматичного керування системами охолодження необхідно мати точну математичну модель динаміки перехідних теплових процесів. Для запобігання цьому недостатньо лише апаратних рішень; необхідна розробка системного програмного забезпечення. Особливої актуальності набуває перехід від реактивного керування охолодженням до предиктивного обслуговування (Predictive Maintenance). Для створення таких алгоритмів необхідна точна математична модель динаміки перехідних теплових процесів з її подальшим втіленням в програмний код [3].

Результати дослідження

Для аналізу динаміки охолодження процесора після зняття обчислювального навантаження доцільно використати модель на основі закону Ньютона [1]. Оскільки у фазі пасивного вистигання активне тепловиділення кристала практично відсутнє ($P \approx 0$), швидкість зниження температури описується лінійним диференціальним рівнянням:

$$\frac{dT}{dt} = -k(T - T_c)$$

де T – поточна температура процесора; T_c – температура навколишнього середовища (повітря всередині корпусу); k – коефіцієнт пропорційності охолодження; t – час.

Для знаходження функції зміни температури від часу застосуємо метод відокремлення змінних [2]. Інтегруючи обидві частини та використовуючи початкову умову, отримуємо розв'язок:

$$T(t) = T_c + (T_0 - T_c)e^{-kt}.$$

Для оцінки ефективності систем охолодження знайдемо час, необхідний для зниження температури процесора з критичної позначки $T_0 = 96^\circ\text{C}$ до цільової безпечної температури $T_{safe} = 55^\circ\text{C}$. Температуру середовища всередині серверної стійки або корпусу візьмемо як сталу $T_c = 32^\circ\text{C}$.

Виведемо формулу для обчислення часу t :

$$55 = 32 + (96 - 32)e^{-kt}$$

$$e^{-kt} \approx 0.359$$

$$t = \frac{-\ln(0.359)}{k} \approx \frac{1.024}{k}.$$

Розглянемо два класичні інженерні сценарії. Для моделювання теплової динаміки процесора прийнято узагальнені значення коефіцієнта охолодження k : для повітряних систем $k = 0.015 \text{ c}^{-1}$, для рідинних систем типу АІО 240 mm $k = 0.06 \text{ c}^{-1}$. Значення підібрані як усереднена апроксимація для сучасних процесорів з TDP 65–170 W і відображають різницю швидкості теплового відгуку систем охолодження [1].

1. Стандартне повітряне охолодження (Tower Cooler). Час охолодження:

$$t_1 = \frac{1.024}{0.015} \approx 68 \text{ секунд.}$$

2. Рідинне охолодження (AIO Liquid Cooler 240mm). Час охолодження:

$$t_2 = \frac{1.024}{0.06} \approx 17 \text{ секунд.}$$

З інженерної точки зору, коефіцієнт тепловіддачі k не є сталою величиною. Внаслідок деградації термоінтерфейсів (пересихання термопасти) та накопичення пилу на радіаторах ефективність розсіювання тепла поступово знижується. Для програмного моніторингу цього фізичного стану розроблено алгоритм, що базується на вираженні поточного коефіцієнта k з отриманого вище розв'язку:

$$k = -\frac{1}{t} \ln\left(\frac{T(t) - T_c}{T_0 - T_c}\right)$$

Розроблено прототип фоновий системної утиліти мовою C++, яка реалізує концепцію предиктивного обслуговування (Predictive Maintenance) [3]. Логіка роботи полягає у фіксації пікової температури T_0 у момент зняття навантаження, вичікуванні системної затримки t та зчитуванні поточного значення $T(t)$ для обчислення деградації системи.

Програмна реалізація математичного ядра алгоритму має такий вигляд:

```
#include <iostream>
#include <cmath>
#include <chrono>
#include <thread>
#include <Windows.h>

double getSimulatedTemperature(bool isPeak) {
    if (isPeak) return 96.0;
    return 72.0;
}

int main() {

    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
```

```

double T_c = 32.0;
double k_ref = 0.015;
int t_delay = 30;

double T_0 = getSimulatedTemperature(true);
std::this_thread::sleep_for(std::chrono::seconds(t_delay));
double T_current = getSimulatedTemperature(false);

double log_argument = (T_current - T_c) / (T_0 - T_c);
double k_current = -(1.0 / t_delay) * log(log_argument);

double degradation = (1.0 - (k_current / k_ref)) * 100.0;

if (degradation > 15.0) {
    std::cout << "УВАГА: Ефективність охолодження впала на "
        << round(degradation) << "%." << std::endl;
    std::cout << "Рекомендовано технічне обслуговування." << std::endl;
}
return 0;
}

```

Важливо зазначити, що у наведеному прототипі температурні показники (T_0 та поточна температура) задані як константи за допомогою функції-заглушки. Це зроблено виключно для верифікації математичної моделі, оскільки прямий доступ до апаратних термодатчиків (Digital Thermal Sensors) на рівні користувача часто обмежується ядром операційної системи з міркувань безпеки.

У повноцінному системному програмному забезпеченні ці константи будуть замінені на функції низькорівневого зчитування даних:

- В ОС Linux: через безпосереднє звернення до системних текстових файлів у віртуальній директорії `/sys/class/thermal/`.
- У середовищі Windows: за допомогою інтерфейсу WMI (Windows Management Instrumentation) або інтеграції з низькорівневими драйверами материнської плати.

Такий підхід алгоритмізує математичну модель Ньютона, перетворюючи пасивний моніторинг температур на активну систему предиктивної діагностики, яка здатна завчасно сигналізувати про фізичний знос охолоджувальних компонентів до настання критичного тротлінгу.

Висновки

Проведене дослідження підтверджує, що для забезпечення стабільності сучасних обчислювальних систем класичне математичне моделювання має супроводжуватися розробкою спеціалізованого системного програмного забезпечення. Використання диференціального рівняння, побудованого на основі закону Ньютона про охолодження, дозволяє з високою точністю прогнозувати динаміку теплових процесів у комп'ютерних системах. Теоретичні розрахунки показали, що системи рідинного охолодження здатні скоротити час перебування кристала процесора в зоні температурного ризику в чотири рази порівняно зі стандартними повітряними кулерами.

Отримана обернена математична функція для визначення коефіцієнта тепловіддачі є обчислювально простою та не потребує значних апаратних ресурсів. Це дає можливість реалізувати її у вигляді низькорівневого коду мовами C або C++ для мікроконтролерів, вбудованих систем та системних утиліт без створення додаткового навантаження на центральний процесор.

Розроблений прототип програмного забезпечення мовою C++ підтвердив можливість переходу від реактивного керування системою охолодження до проактивного моніторингу її технічного стану. Запропонований алгоритм здатний у фоновому режимі виявляти деградацію термоінтерфейсу та забруднення радіатора, що призводять до зниження ефективності тепловідведення на 15 % і більше, та своєчасно сповіщати користувача ще до виникнення критичного теплового тротлінгу.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Incropera F. P., DeWitt D. P. Fundamentals of Heat and Mass Transfer. 7th ed. – Hoboken : John Wiley & Sons, 2011. – 1048 p.
2. Boyce W. E., DiPrima R. C. Elementary Differential Equations and Boundary Value Problems. 10th ed. – Hoboken : Wiley, 2012. – 640 p.
3. Mobley R. K. An Introduction to Predictive Maintenance. 2nd ed. – Oxford : Butterworth-Heinemann, 2002. – 438 p.

Лисий Костянтин Андрійович – студент групи 2ПІ-256, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: garoshh1818@gmail.com.

Науковий керівник: **Прозор Олена Петрівна** – к.пед.н., доцент кафедри вищої математики, Вінницький національний технічний університет, Вінниця, email: prozor@vntu.edu.ua.

Lysyi Kostiantyn A. — Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, email : garoshh1818@gmail.com.

Scientific supervisor: **Prozor Olena P.** – PhD (in Pedagogical Sciences), Docent, Associate Professor of the Department of Higher Mathematics, Vinnytsia National Technical University, Vinnytsia, email: prozor@vntu.edu.ua.