Thermal Equilibrium Simulation for CPU Heatsink

BOUJIDA Nezar

18/11/2023

The simulations discussed in this paper were initially developed by Mr. Charles Bouillaguet. Replicating them was essential to grasp the concepts detailed in the accompanying paper on parallelization, which focuses on optimizing these simulations.

1 About Simulation

The simulation models the transfer of thermal energy between two contacting bodies, such as a CPU and its heatsink, until equilibrium is reached.
It encompasses the calculation of heat flow, based on material properties like mass thermal capacity, and the application of four distinct heat transfer types: conduction, convection, radiation, and within-material propagation.

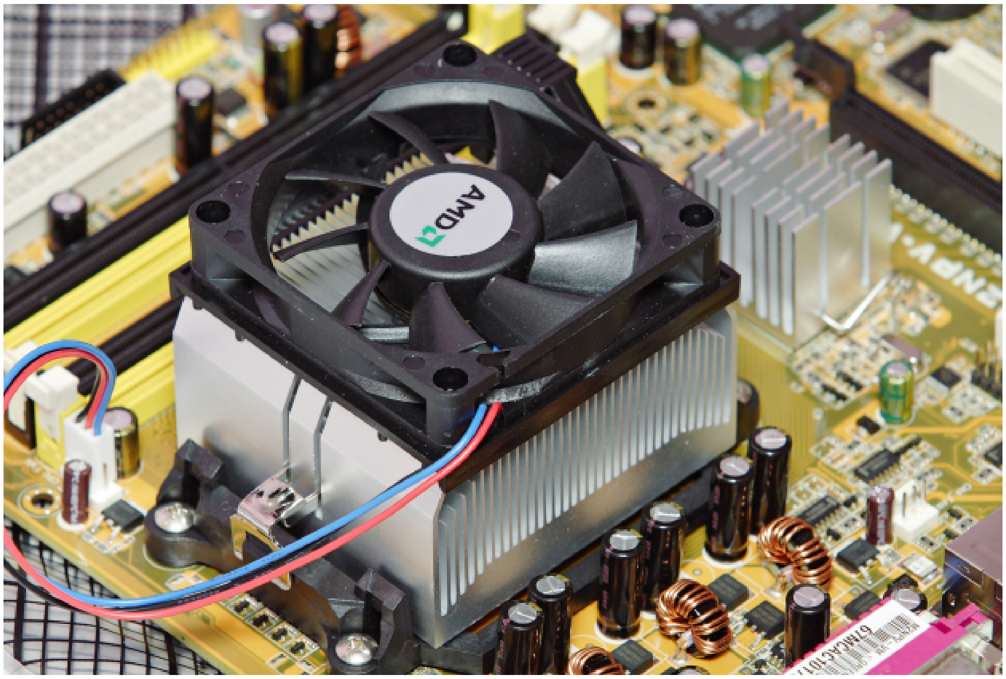

Figure 1: A fan-cooled heat sink on a processor.

2 Simulation Context and Hypothesis

Firstly, we hypothesize a constant medium temperature, implying that the surrounding environment's temperature (whether air or coolant) where the heatsink functions remains uniform and unchanging over time.
Additionally, we disregard external thermal energy gains by the heatsink through radiation, indicating that the heatsink does not significantly absorb thermal energy from external sources such as sunlight or nearby heat sources.

2.1 Spatial Discretization

Spatial discretization involves converting the heatsink's continuous space into a finite 3D mesh of cuboidal cells, accurately reflecting its physical structure for manageable equation resolution. This approach standardizes cell shape to facilitate heat flux calculations and allows for tuning simulation accuracy by adjusting mesh size, with finer meshes yielding more precise results at the expense of computational resources.

2.2 Temporal Discretization

Temporal discretization complements spatial discretization by segmenting the simulation's duration into discrete steps. This enables the model to track and calculate temperature changes incrementally, offering a comprehensive framework for systematically analyzing thermal dynamics.

2.3 Memory Representation

To represent the three-dimensional temperature array T in memory for dimensions $n \times m \times o$ (corresponding to the axes x, y, z), we employ a row-major order, a common convention in the C language. This approach stores the xy planes contiguously in memory. Specifically, for a cell at coordinates (i, j, k), its temperature is stored at $T[u]$, where:

- $T[u + 1]$ represents the temperature of the neighboring cell along the x-axis $(i + 1, j, k)$
- $T[u + n]$ for the next cell along the y-axis $(i, j + 1, k)$
- $T[u + nm]$ for the adjacent cell along the z-axis $(i, j, k + 1)$.

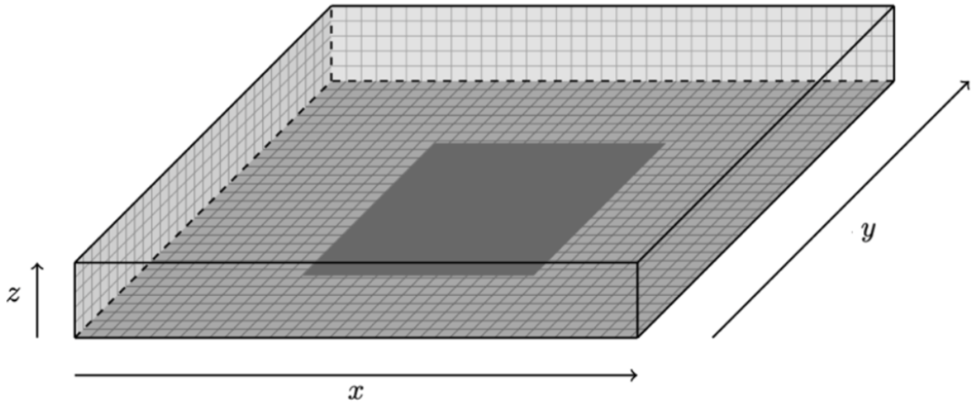

Figure 2: Schematic illustration of the heat sink.

3 Implementation

The simulation involves iterating over each cell, updating temperatures based on the different types of heat transfer, and checking for convergence when the temperature variation falls below a certain threshold, indicating that equilibrium is nearly reached.

The updating of the temperature involves calculating the net heat flux received by each cell and then determining how this affects the cell's temperature over a given time step. This process, which is central to the simulation, incorporates the various modes of heat transfer: conduction, convection, and radiation.

```
procedure SIMULATION(...)
    Allocate two arrays $R$ and $T$ of size $nmo$
    $T[:] \leftarrow 293.15$                                ▷ all cells are initially at 20 °C
    $t \leftarrow 0$
    convergence $\leftarrow 0$
    while convergence $= 0$ do
        for $0 \leq k < o$ do                               ▷ Processes the $k$th $xy$-plane
            for $0 \leq j < m$ do
                for $0 \leq i < n$ do
                    $R[knm + jn + i] \leftarrow$ UPDATETEMP$(i, j, k)$
                                                            ▷ Accesses to up to 6 neighbor cells
                end for
            end for
        end for
        if $t$ is an integer then
            $T_{max} = \max R[:]$
            $\varepsilon = \sum_{u=0}^{omn} (R[u] - T[u])^2$
            if $\sqrt{\varepsilon}/\Delta_t < 0.1$ then
                convergence $\leftarrow 1$
            end if
            Print $t$ and $T_{max}$ to keep the user waiting...
        end if
        $t \leftarrow t + \Delta_t$
        $T \leftarrow R$
    end while
    Save $T$ in a file for the forthcoming graphical rendering
end procedure
```

Figure 3: Principle of the simulation.

3.1 Understanding Heat Flux Contributions

In the thermal simulation of a heatsink, heat transfer occurs through conduction, convection, and radiation.

Conduction is modeled inside the heatsink, where heat moves from hotter to cooler parts according to Fourier's law, taking into account the thermal gradient and applying specific heat flux density for cells in direct contact with the CPU.

Convection is calculated for cells at the heatsink's surface, with the convective heat flux determined by the temperature difference between the cell surface and ambient air, using the convection heat transfer coefficient.

Radiation loss from surface cells is estimated using the Stefan-Boltzmann law, based on the cell's temperature and the assumption that it behaves like a black body, completing the comprehensive approach to simulating heat dissipation mechanisms in the heatsink.

3.2 Temperature Update Mechanism

In the simulation, heat flux for each cell is calculated by accounting for incoming and outgoing heat through conduction, convection, and radiation, factoring in the cell's current temperature, adjacent cells' temperatures, and environmental conditions.

The net heat flux (Φ) determines the temperature change (ΔT), calculated from the cell's mass, specific heat capacity (c), time step (Δt), material density (ρ), and cell dimensions (Δx, Δy, Δz).

Temperatures are updated iteratively at each time step based on ΔT, pushing the system towards thermal equilibrium.

The code for the simulation can be found in the file named `sequential.c`.

4 Foundational Concepts for Thermal Simulation

This section outlines the fundamental concepts and requirements that underpin the thermal simulation.

4.1 Heat Flow and Flux

When two bodies come into contact, they exchange thermal energy until they reach the same temperature. This exchange is quantified as heat flow (in Watts) and heat flux density, which is the rate of energy transfer per unit area. For example, the simulation considers a processor with a specific contact surface area with its heatsink, calculating the heat flux density based on this area.

4.2 Material Properties

Key to understanding how materials react to thermal energy is their mass thermal capacity, indicating how much energy is needed to raise the temperature of a unit mass by one degree. This property is critical for simulating how heat spreads within materials.

4.3 Fourier's Law Basics

Principle: Fourier's Law states that the rate of heat transfer (Q, in watts) through a material is proportional to the negative gradient of temperatures ($-\nabla T$) and the area (A, in square meters) perpendicular to the direction of heat transfer. The proportionality constant is known as the thermal conductivity (λ, in watts per meter-kelvin), a material-specific property that quantifies its ability to conduct heat.

Mathematical Expression: The mathematical expression of Fourier's Law in one dimension is:

$$Q = -\lambda A \frac{dT}{dx}$$

where $\frac{dT}{dx}$ represents the temperature gradient in the direction of heat flow.

4.4 Principles of the Stefan-Boltzmann Law

Foundational Concept: This law states that the total energy radiated per unit surface area of a black body per unit time (J, in watts per square meter) is directly proportional to the fourth power of the black body's absolute temperature (T, in kelvins).

Mathematical Expression: The law is mathematically represented as

$$J = \sigma T^4$$

where σ is the Stefan-Boltzmann constant.

4.4.1 Application in Heatsink Simulations

Radiative Heat Loss: In the context of heatsink design and thermal management, the Stefan-Boltzmann Law is applied to estimate the radiative heat loss from the heatsink's surface. Since heatsinks aim to dissipate heat from components like CPUs, understanding and optimizing for radiative heat loss is crucial for enhancing cooling efficiency.

Black Body Assumption: For simplicity, simulations often assume the heatsink behaves as a perfect black body, which is an ideal emitter and absorber of thermal radiation. While real materials may not achieve perfect black body radiation, this assumption simplifies calculations and can be adjusted with emissivity factors for more accuracy.