Real-time system feedback thermal control on multicore processors

INSTRUCTOR: Prof. Indranil Saha

TEAM:

Kaushik Raj V Nadar (200499)

Napa Jaya Prakash (200622)

Prateek Kumar Pandey (200710)

CS637 PROJECT

CONTENTS

Problem

Setup

Model

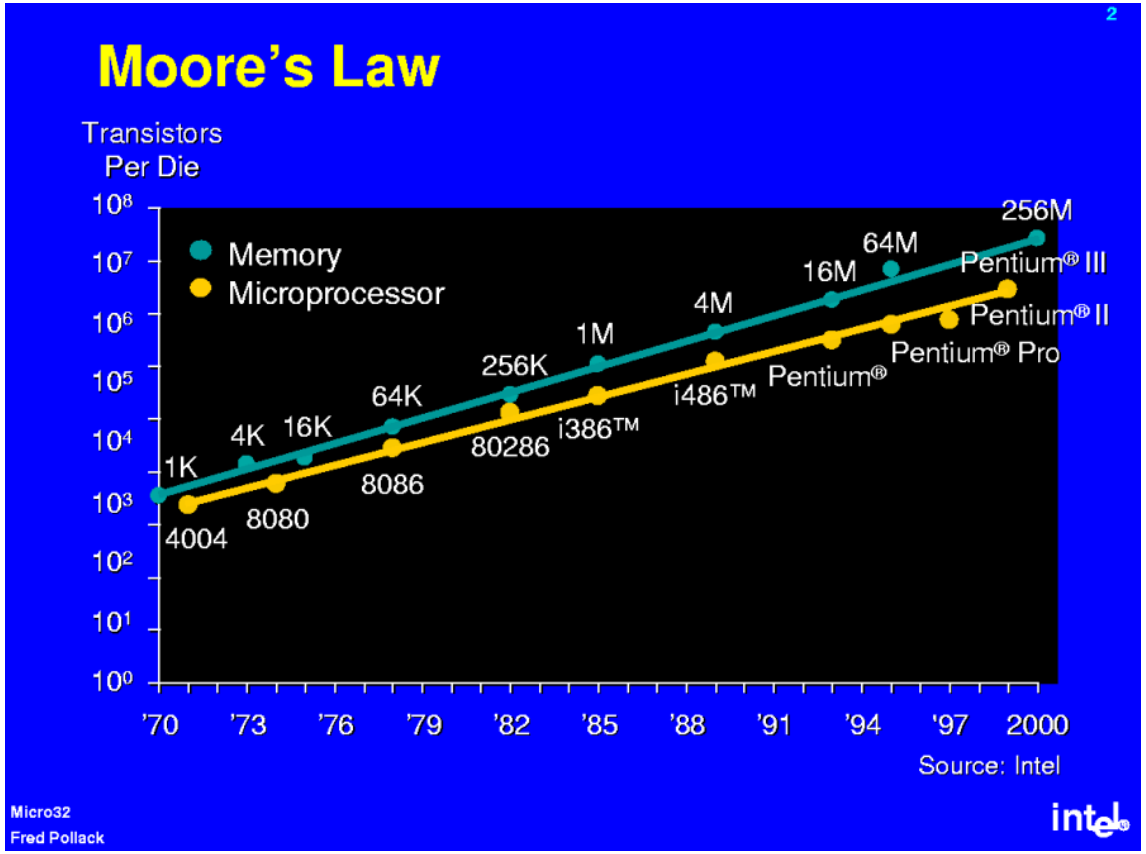
Results

PROBLEM

All electronic devices and circuitry generate excess heat and thus require thermal management to improve reliability and prevent premature failure. Therefore, thermal management of embedded systems is necessary either through external cooling or software optimizations.

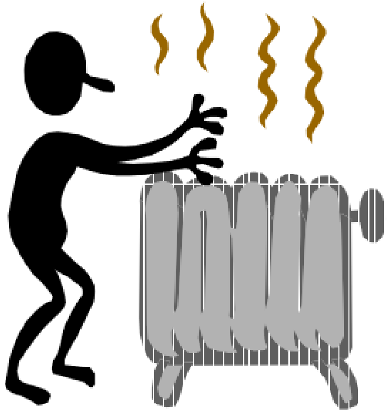
Moore’s Law :

- The Good News: 2X Transistor counts every 18 months
- The Bad News: To get the performance improvements we’re accustomed to, CPU Power consumption will increase exponentially too…



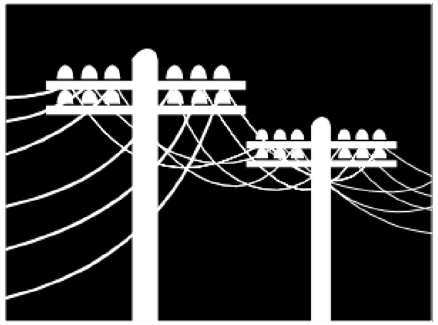
WHY WORRY ABOUT HEAT DISSIPATION?

Quality of Service



Thermal issues: affect cooling, packaging, reliability, timing

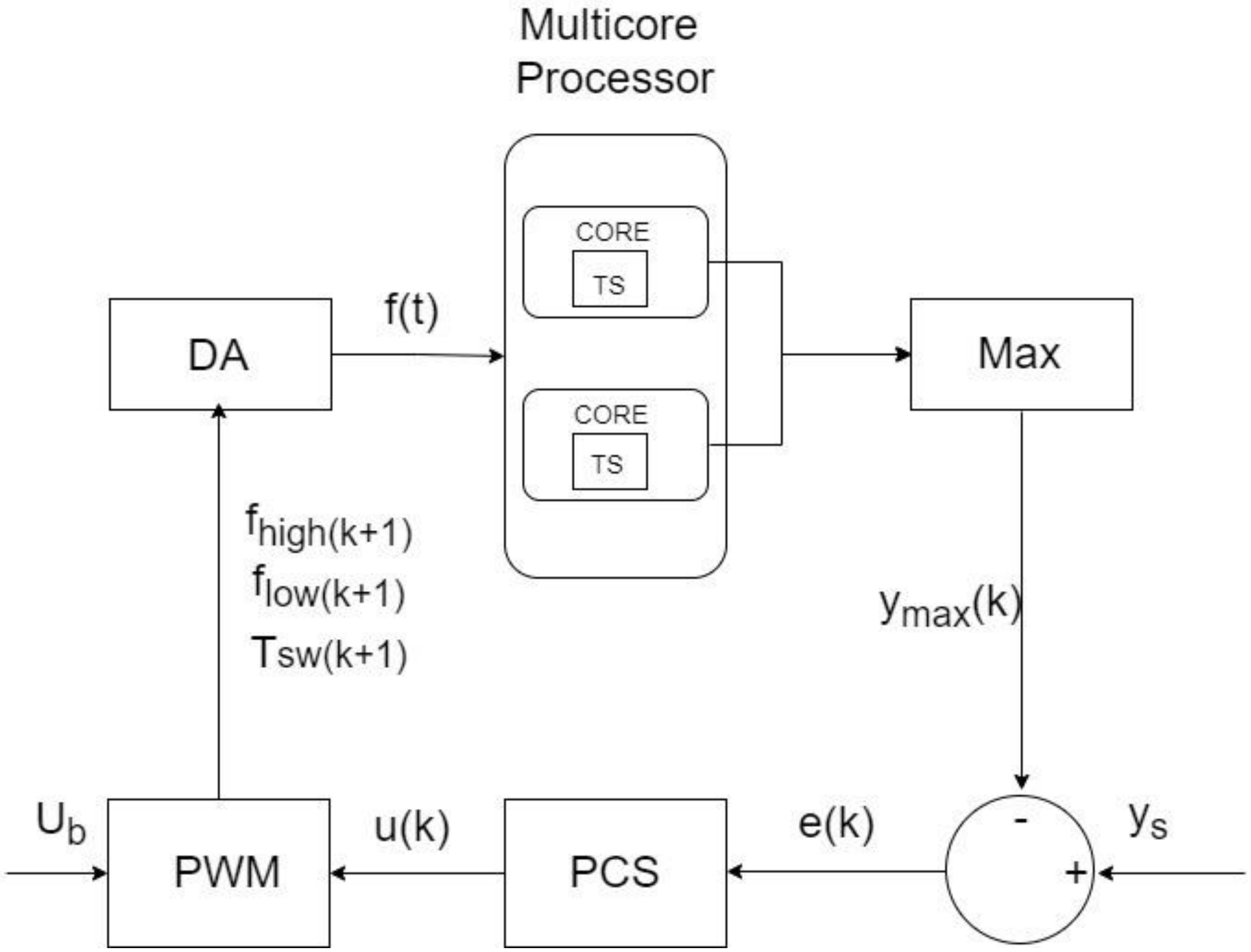
Environment



SETUP

- The CPU is assumed to provide Dynamic Voltage and Frequency Scaling (DVFS). We assume that DVFS in prevalent multicore CPUs have two properties. First, the frequency and voltage of all cores can only be scaled evenly, i.e., they must always be identical. Second, the CPU only supports discrete frequency range.
- We assume tasks on a core are scheduled locally using a real-time scheduling policy with a known schedulable utilization constraint U_b, such as Rate Monotonic (RM) or Earliest Deadline First (EDF) under particular circumstances. The tasks on a core l meet their deadlines if $U_l \leq U_b$ where U_l is utilization of individual core l.
- Given a multicore embedded real-time system, our objective is to manage the processor temperature such that the maximum temperature across all cores follows a temperature set point, y_s, subject to the restriction of utilization bound U_b on each processor core. y_s is the intended processor-tolerable temperature.

FEEDBACK CONTROL LOOP OF THE MODEL



PROPORTIONAL CONTROLLER AND SATURATION (PCS)

- The PCS computes the controller output u(k) as follows:

$$u(k) = \begin{cases} 1, \text{ if } k_p e(k) > 1, \\ -1, \text{ if } k_p e(k) < -1, \\ k_p e(k), \text{ otherwise;} \end{cases}$$

where k_p is the coefficient of proportional control and $e(k) = y_s - y_{max}(k)$. The output of the controller is limited to the range $[-1, 1]$

- The PCS is designed based on passivity and can accommodate the nonlinearities induced by the Max function and the saturation. There are various precise mathematical definitions for passive systems that essentially state that the output energy must be bounded so that the system does not produce more energy than was initially stored. Under certain technical conditions, strictly input and strictly output passive systems are Lyapunov stable. In this case, passivity offers advantages for computing a Lyapunov function that is used to prove stability of the closed-loop system.

PULSE WIDTH MODULATION (PWM)

The continuous input to the PWM in the k^{th} sampling period is $u(k) \in [-1, 1]$. The PWM computes ($f_{high}(k + 1)$, $f_{low}(k + 1)$, $T_{sw}(k + 1)$) based on u(k). The upper limit of the output corresponds to the maximum frequency supported by the processor. The lower limit of the output corresponds to the lowest frequency that satisfies the utilization bound or the minimum frequency, whichever is higher.

PWM initially selects two successive frequency levels f_i and f_{i+1} from the discrete frequency set that fulfil $f_i < f_u(k) < f_{i+1}$ to minimize CPU speed change where $f_u(k) = f_{min} + (f_{max} - f_{min}) \frac{u(k)+1}{2}$

The time to switch from $f_{high}(k + 1)$ to $f_{low}(k + 1)$ is computed as

$$T_{sw} = \frac{f_u(k) - f_{low}(k+1)}{f_{high}(k+1) - f_{low}(k+1)} T_s,$$

Where T_s is the sampling period.

CPU POWER MODEL

Average Power of each core is given by :

$$\bar{P}(k) = \bar{P}_a(k) + C_y y(k)$$

where $P_a(k) = U(k)C_2V^3(k) + C_0(V(k))V(k)$ and $C_y = C_1(V(k))$. $P_a(k)$ and C_y can be expressed in terms of the frequency, based on the relationship between supply voltage and frequency, $V(k) = K\ f(k) + V_{th}$ and $\frac{U(k)}{f(k)} = \frac{U_0}{f_0}$ where U_0 and f_0 are the initial CPU utilization and frequency.

After incorporating the PWM :

$P_{a,max}$ and $P_{a,min}$ are the average power consumption at f_{max} and f_{min}, respectively. We can add PWM to the power model using

$$\bar{P}(k) = G_p(P_{ap}u(k) + P_{am}) + C_y y(k)$$

where $P_{ap} = (P_{a,max}−P_{a,min})/2$, $P_{am} = (P_{a,max}+P_{a,min})/2$, and G_p is the gain to represent the uncertainty caused by power variation

CPU THERMAL MODEL

Temperature dynamics of the CPU is given by :

$$\dot{\mathbf{Y}}(t) = A\mathbf{Y}(t) + B_P\mathbf{P}(t) + B_y y_0$$

Where Y(t) is the temperature vector of the CPU cores, P(t) is the Power vector, and y_0 is the ambient temperature.

The parameters A, $B_{p,}$ and B_y depend on the thermal resistances and the capacitances of the cores.

The ZOH (Zero Order Hold) equivalent :

If power is held constant and the average environmental temperature is $y_0(k) = \frac{1}{T_s} \int_{(k)T_s}^{(k+1)T_s} y_0(t)dt$ during the kth sampling period.

$$\mathbf{Y}(k+1) = \Phi_o\mathbf{Y}(k) + \Psi_P\bar{\mathbf{P}}(k) + \Psi_y y_0(k)$$

where $\Phi_o = e^{AT_s}$, $\Psi_P = (\int_0^{T_s} e^{A\tau}\, d\tau)B_P$, $\Psi y = (\int_0^{T_s} e^{A\tau}\, d\tau)B_y$ and $\mathbf{P}(k)=[P_1(k),\ldots, P_M(k)]^T \in R^M$.

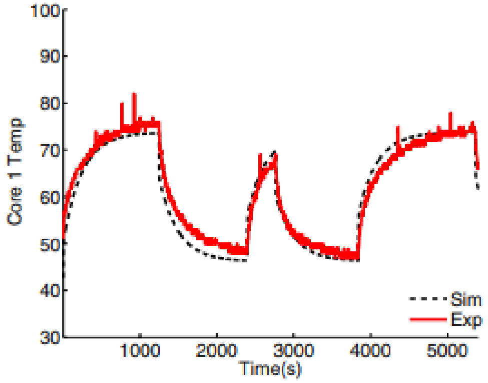
Substituting the power model here results in

$$\mathbf{Y}(k+1) = \Phi\mathbf{Y}(k) + P_{ap}\Psi_P G_p u(k) + \Psi_y y_0(k) + P_{am}\Psi_P G_p$$

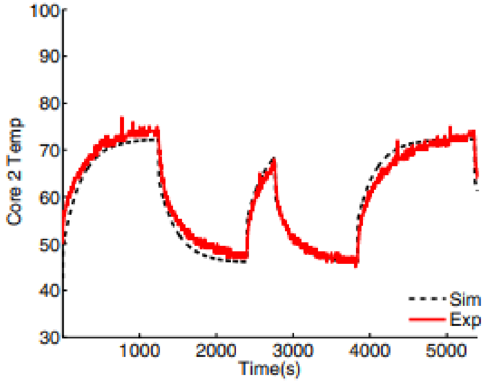
in which $\Phi = (\Phi_o + C_y\Psi_P\,[\ I_M\ 0\ ])$ where $I_M \in R^{M\times M}$ denotes the identity matrix.

RESULTS

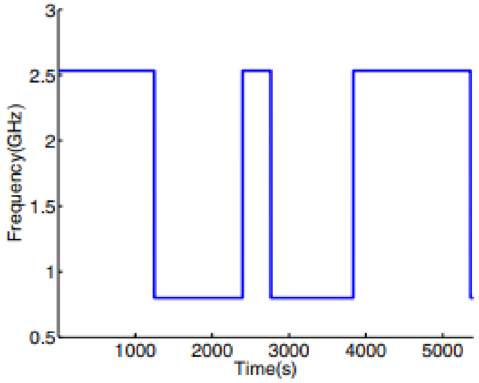
CPU Simulation



(a) Core 1 Temperature

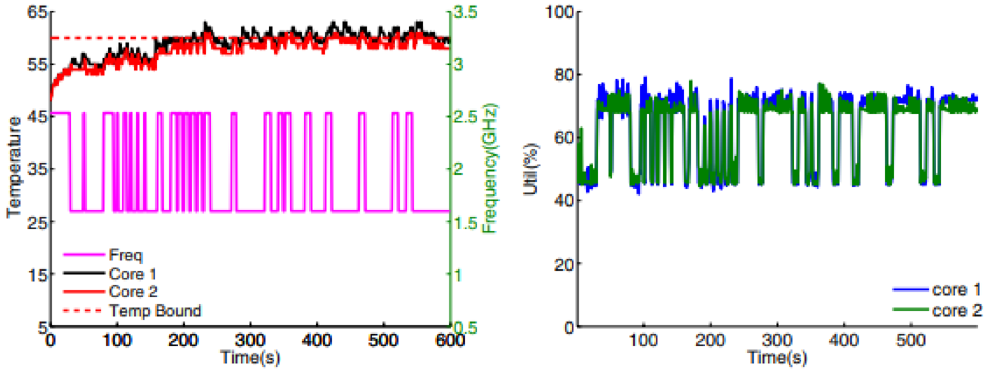


(b) Core 2 Temperature

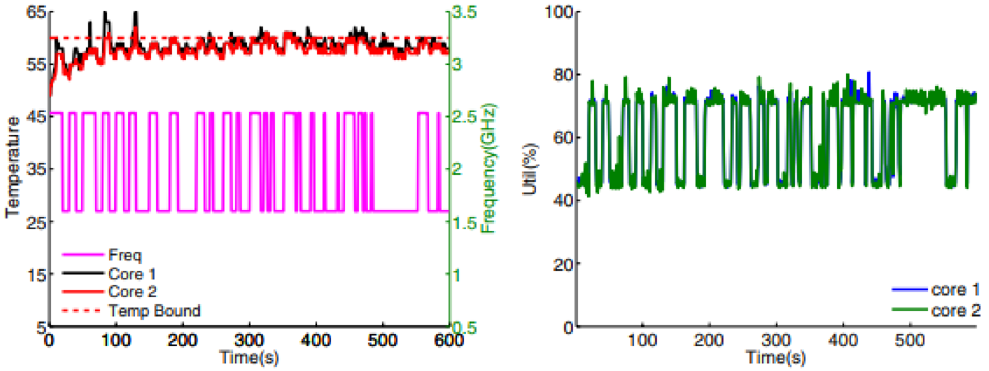


(c) Frequency

Simulation during workload

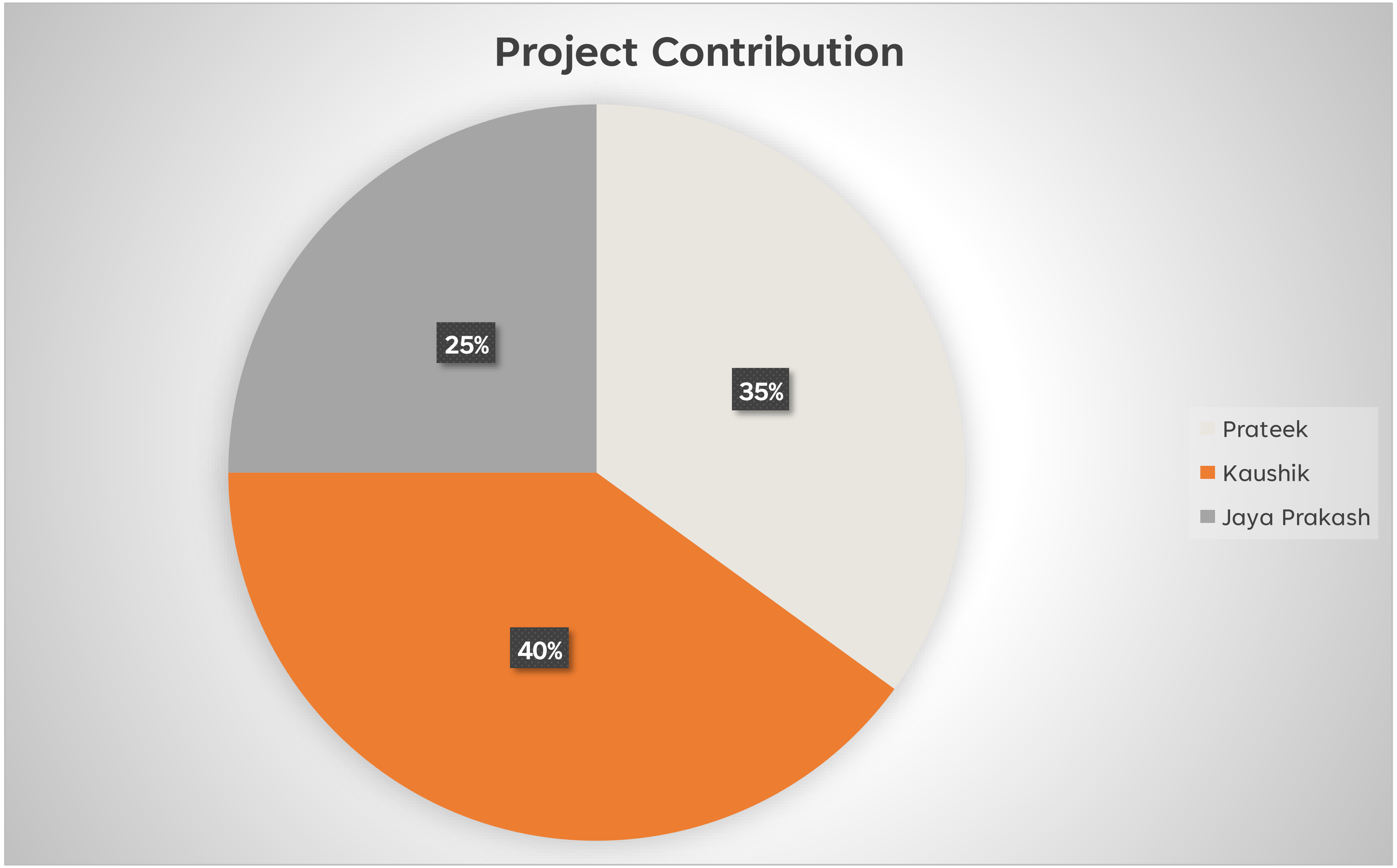


(a) Mixed Workload



(b) CRC

MEMBER CONTRIBUTION



CONCLUSION

We tried to implement the Thermal management system proposed in this paper.

As of now we have implemented the PCS, PWM, Max Blocks of the model.

The implementation of CPU model is incomplete as we couldn’t find adequate resources for modeling CPU’s thermal dynamics. Hence, we couldn’t do our own simulations.

THANK YOU