Optimization and Algorithms

Temperature control of multi-core platforms

André Pereira Guilherme Pires João Lages
andremfp29@gmail.com gpires@tutanota.com joaolages@tuta.io
Instituto Superior Técnico

***Abstract*—Multi-core chips are prone to reach high temperatures, possibly exceeding safety limits. In fact, if the temperature iks left uncontrolled, it can surpass a critical threshold and make the chip stop working. In this project, we use convex optimization to control the frequency of a 5x2 core grid in four different scenarios, attempting to maximize it, without exceeding the temperature threshold.**

I. Introduction

In this work, we maximize the operating frequency of each individual core in a 5x2 grid architecture, subject to temperature constraints.

Chips with multi-core architectures can easily reach high operating temperatures, not only due to the high frequencies at which they (desirably) work but also due to heat conduction between adjacent cores.

In general, one wants the cores to operate with the highest frequency possible. However, too high of a temperature can damage the chip and prevent it from working properly. Given the model that describes the evolution of each core's temperature over time, as a function of its previous temperature and operating frequency, it is possible to calculate, at each time instant the best (maximum, in this case) possible temperature for each core. Despite the model's simplicity, the number of interactions at play makes these calculations not ammenable to be manually computed. Moreover, even a skilled individual would have trouble computing these temperatures for a decent number of time steps, quickly enough for the result to be useful.

That being said, this paper shows how to use convex optimization to solve this problem in an acceptable amount of time, using an open-source general convex optimization toolbox.

The remainder of the paper is organized as follows. In Section II, we introduce a general description of this problem as an optimization problem. We then divide the work in two parts, each composed of different subproblems that derive from this general description, each subject to different special constraints. Section III Explains the strategies employed in solving each of the subproblems. The Numerical results of each of the experiments are presented in Section IV. Finally, we state our conclusions in Section V.

II. Problem formulation

The general optimization this work seeks to solve is:

$$\begin{array}{ll} \text{maximize} & \sum_{i,k} f_{i,k} \ , \ \forall i,k \\ \text{subject to} & t_{0,i} = t_{start,i} \ , \ \forall i \\ & t_{k+1,i} = t_{k,i} + \sum_{\forall j \in Adj_i} a_{ij}(t_{k,j} - t_{k,i}) + b_i p_{k,i} \ , \ \forall k,i \\ & t_{k,i} \leq t_{max} \ , \ \forall k,i \\ & p_{max}(\frac{f_i}{f_{max}})^2 \leq p_{k,i} \ , \ \forall k,i \\ & f_{k,i} \geq 0 \ , \ \forall k,i \end{array} \tag{1}$$

.

Where the variables subscripted with *i* refer to the *i*-th core and the ones subscripted with *k* refer to the *k*-th time step; and *t* stands for temperature, *p* stands for power, *f* stands for frequency, Adj_i stands for the set of cores adjacent to to the *i*-th core, f_{max} stands for the maximum frequency of this core architecture, t_{max} stands for the maximum temperature the cores are allowed to achieve, a_{ij} stands for a constant that describes the heat conductivity between adjacent cores and b_i stands for a constant that describes how much each core's power contributes for its temperature. The optimization variables are $f_{k,i}$, *i.e.*, each core's frequency, at each time step. Note that we don't solve an optimization problem per time step, but rather one optimization problem that is 'aware' of the time interval during which the cores will be operating, so when we say "maximize each core's frequency at each time step" we mean "maximize each core's frequency at each time step, knowing that the cores will be operating for a given amount of time and so we can't simply set f = fmax at each time step".

In the next two subsections, we'll describe, in the first one, a set of problems that can be tackled by tweaking this formulation (specificaly, by adding constraints to it); and in the second one, a single but more complex problem, that seems tricky to solve at first glance, but can be easily solved with the appropriate formulation.

A. Part 1

In the first part, we proposed to formulate and solve the following problems:

1) Maximize each core's frequency at each time step (simply applying the original problem formulation)
2) Maximize each core's frequency at each time step, under the influence of an external heat conductor (whose behaviour is previously known, *i.e.*, the magnitude with which it conducts heat per time instant is previously

known. The equation that governs the evolution of temperature becomes:

$$\begin{aligned} t_{k+1,i} = t_{k,i} + \textstyle\sum_{\forall j \in Adj_i} a_{ij}(t_{k,j} - t_{k,i}) \\ +b_i p_{k,i} + \tfrac{a_{ij}}{10} t_{k-1,ext} \ , \ \forall k, i \end{aligned} \tag{2}$$

, where $t_{k,ext}$ stands for the temperature of the external conductor, at time k.

3) Maximize each core's frequency at each time step, under the influence of an external heat conductor, as described in the previous item, but now minimizing frequency transitions. The relevance of this is due to the fact that, in a real case scenario, having a core altering its frequency with a high rate is undesirable since it results in an unstable performance and possibly additional power dissipation in the form of heat. The problem "objective" becomes:

$$\text{minimize} \quad \textstyle\sum_{i,k} -f_{k,i} + \rho \|f_{k,i} - f_{k-1,i}\| \ , \ \forall i, k \tag{3}$$

.

4) Lastly, we run the previous problem several times giving different weights to the frequencies sum and the frequency transitions components of the cost function. We plot the results to obtain a trade-off curve between maximizing the frequencies and having a little amount of frequency transitions.

B. Part 2

In the second part, we consider a case in which the *b* parameters (that describe how each core's power influences its temperature) are uncertain (within a set of 3 possible configurations). That being the case, we wanted to find the best possible frequency assignment for each time step, that doesn't violate the temperature constraints if applied to any of the 3 possible configurations.

III. Approach

A. Part 1

In a first instance, there was the need to find fixed values for the following variables: *k*, *b* and initial temperatures for each core.

Since we defined t_{max} as $100°C$, the initial temperatures for each core were fixed in something between 40 and 50 degrees, so we could see a proper evolution of the system. For the time horizon we chose a value high enough for the temperatures to stabilize, when influenced only by the multiple $a_{ij}s$.

The formulation of the optimization problem was then constructed as mentioned in the previous section, without the constraint of the maximum temperature. With this formulation, the values of *b* were obtained so that the final temperatures would be over t_{max}. After fixing these *b* values, the constraints of the maximum temperature for each core were added to the formulation.

1) The whole problem described above was implemented in python, using CVXPY [2], a package that resolves convex problems. The variables of the problem are the frequencies, powers and temperatures of each core in each time step.
2) When the influence of an external heat source was considered, a new component was added to the temperature of each core in a specific time horizon (between $\frac{1}{3}k$ and $\frac{2}{3}k$), like it was explained in (2).
3) The same procedure was used for minimizing frequency transitions, except that in this case the objective of the problem changed like it was described in (3). The value of p was randomly chosen until it produced significant differences in the previous results.
4) The trade-off curve was then obtained by running multiple versions of this problem with different p values.

B. Part 2

In the second part, we went back to the first formulation described and twisted it a bit. Two new grids of cores were added to the problem which generated the need to find new values of *b* and initial temperatures for them. There were given similar values to those we obtained previously.

Since the cores followed the same restrictions as before, it was not necessary to make new equations besides the ones described in section II.

The idea was to view all grids working at the same frequency in every time step, having different *b* values. In this case of 3 possible configurations, there will be 7 variables: 3 vectors of powers and temperatures for each grid but only 1 vector of frequencies.

This way, in every time step, all constraints will be considered and the best possible frequency will be chosen accordingly.

IV. Numerical Results

Using matplotlib (a Python package for math plotting), we obtained the following graphs (sorted by the same indexing used in II.

The next graph shows how we came up with the value *k*= 250.

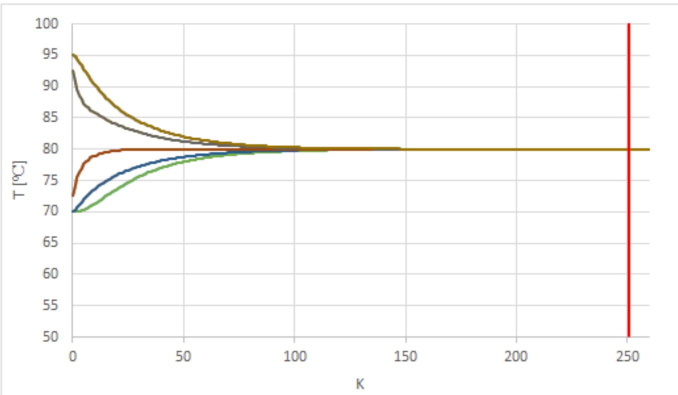

Fig. 1: Choosing a value of *k* high enough for the temperatures to stabilize

The values for *b* were then obtained like it was explained previously (randomly between $0,06$ and 0.1).

A. Part 1

In the first part, we proposed to formulate and solve the following problems:

1) We see the temperatures ”converge” to a straight line and then follow it until the end of the time interval, where they reach the temperature threshold. In the first time steps, the cores that start with colder temperatures have an increase in temperature, while the cores that start with a hotter temperature have a decrease. Considering the Power and Frequency plots, we understand that this happens because the cores that start hotter hold a lower frequency (and thus power) almost until the end, while the cores that start colder hold a higher frequency until the end. In the last time steps, the frequencies (and powers) behave somewhat erracticaly, but in general converging to a value more or less in the middle between the highest and lowest frequencies.

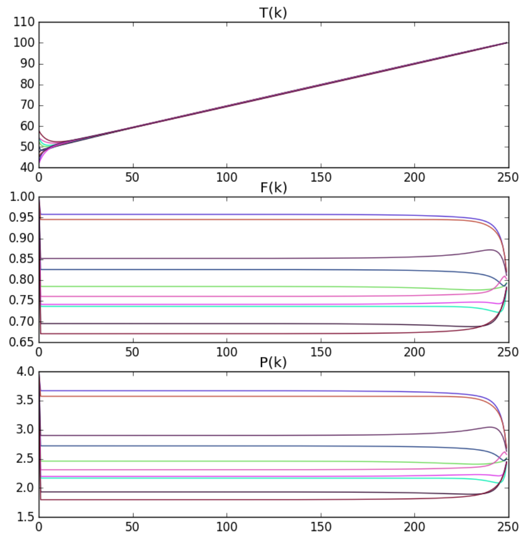

Fig. 2: Maximize each core’s frequency at each time step

2) The results are similar to the previous ones. However the influence of the external conductor is noticeable, because the curve that the temperatures follow in steady state is no longer a straight line, but rather the sum of a straight line with an impulse (which is precisely the behavior of the conductor’s ”heat wave”). As expected, the maximum frequency achieved is lower than in the previous case.

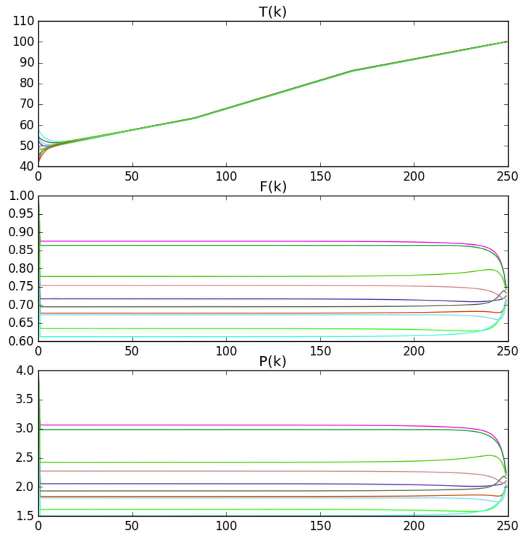

Fig. 3: Maximize each core’s frequency at each time step, under the influence of an external heat source

3) The results are once again similar to previous ones. However, the last time steps, where in the previous cases the frequencies used to behave ”erratically”, in this case the frequency plot has less ”spikes” and progresses more smoothly.

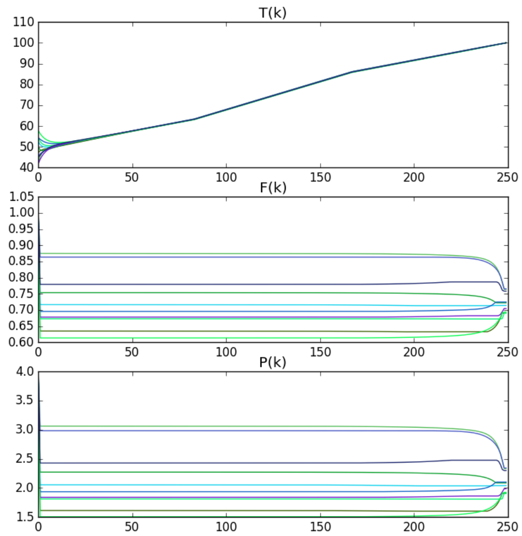

Fig. 4: Maximize each core’s frequency at each time step, under the influence of an external heat source, but now minimizing frequency transitions

4) In this last case, we see a trade-off graph between the two terms in the objective function: the x axis corresponds to the sum of frequencies, the y axis corresponds to the term that accounts for the ”amount of frequency transitions”. The different points are obtained by running the previous experiment with different

values of the weight given to the second term. These increase exponentially in the inverse direction of the x axis, from which we conclude that for a significant increase in the said weight, one doesn't need to have a great penalty in the maximum frequency obtained.

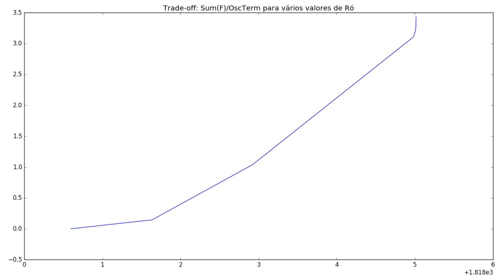

Fig. 5: Trade-off graph between the sum of frequencies and the frequency oscillation term

B. Part 2

For this part, we have to remember that all the grids of cores start with the same initial temperatures and the only thing that differentiates them are the b values, by that means, some core's power contributes higher or lower to its temperature, compared to the same core on another grid.

However, the grids are subject to the same vector of frequencies, that we can see in the next figure.

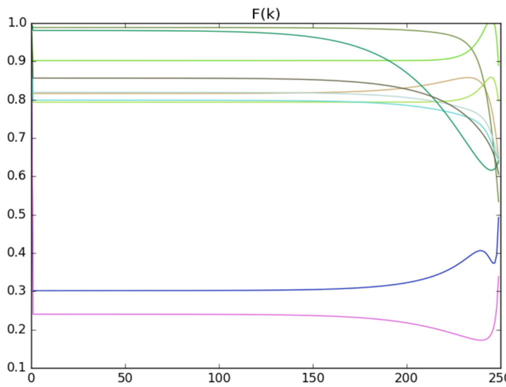

Fig. 6: Frequencies for each core in every core grid along the time horizon

Since the frequencies are identical, the only term that will make a difference in here from the equations stated in (1) is $b_i p_{k,i}$. A temperature of a core is proportional to its b and power values. As it happened before, the objective of the problem is to maximize the final sum of frequencies. In order to achieve this maximum value, CVX looks at a set of constraints for the whole time horizon and chooses values of frequency that imply different values of temperature and power to satisfy our objective. These values of temperature require cores with a higher b to lower its power compared to the cores with a lower value of b and vice-versa.

As we can see in the next images, the temperatures of the grids are really identical, while the differences in the power's vector is more clearly, like it was previously pointed. The similarity of the temperatures is explained due to the fact that all the grids start with the same initial temperatures and it is required to have the maximum temperature (or as close as possible) at the end to achieve our objective of maximizing the sum of frequencies.

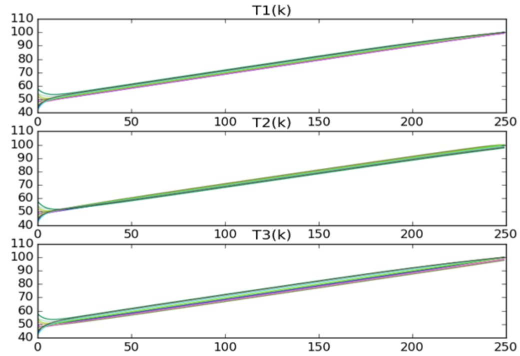

Fig. 7: Temperatures for all the core grids along the time horizon

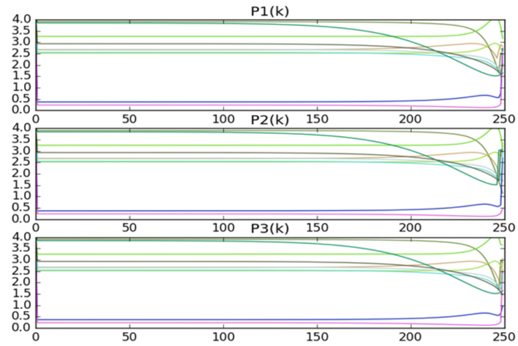

Fig. 8: Powers for all the core grids along the time horizon

V. Conclusions

We conclude that many problems that are intractable for humans are easily solvable with the appropriate problem formulation and optimization techniques and tools. We can observe that the results obtained in this sort of problems, despite being relatively easy to interpret once computed, are completely unobvious for the naked eye, thus asserting the relevance and importance of the methods employed here.

References

[1] Srinivasan Murali and Almir Mutapcic and David Atienza and Rajesh Gupta and Stephen Boyd and Luca Benini and Giovanni De Micheli, *Temperature Control of High-Performance Multi-core Platforms Using Convex Optimization*

[2] Steven Diamond and Stephen Boyd, *CVXPY: A Python-Embedded Modeling Language for Convex Optimization*