Energy Performance Analysis of Advanced Transactional Models

Umut Ucak, Alfredo Capozucca

[a]*University of Luxembourg, 6, avenue de la*
Fonte, Belval, L-4364, Esch-sur-Alzette, Luxembourg

Abstract

Nowadays, most devices come with a multi-core chip. The maximisation of the chip's usage is reached by using software that exploits the benefits of concurrent computing. Modern programming languages provide built-in support for programmers to easily implement the concurrency model of their choice, for example transactions. The ultimate goal of such an implementation is to optimise the time performance of the software as this is often a key property when assessing the quality of a software. However, the cost of real-life operations is not one-dimensional, especially with the ever-approaching energy crisis. It is also important to pay attention and raise awareness about energy consumption rates when pursuing time performant software.

The goal of this study is to analyse advanced transactional models for the purpose of evaluation with respect to their energy consumption. As a transactional model is a solution to deal with concurrency, the analysis brought by this study is meant to find out what is the energy footprint that comes along with concurrency and see if there exists a trade-off between time and energy performance.

A modern cutting-edge transactional model named CHOCOLA is used as reference to benchmark its energy consumption. Since the energy consumption is measured running the same scenarios as in the original work where this advanced transactional model was presented, it is possible to analyse not only its energy performance but also to find out how it relates to time performance.

The collected evidence shows that CHOCOLA is not only more time performant (as initially claimed) but also energy performant with respect to state-of-the-art transactional models. Moreover, evidence has been found that this trend remains the same even under conditions that exceed the

Preprint submitted to Sustainable Computing: Informatics and Systems October 3, 2023

computing capacity of the hardware used for its execution. Another finding this study brings is that performance and energy consumption are correlated when processing transactions. Collected evidence has shown that the correlation is negative when relying on standard transactional models, meaning that higher performance (i.e. less processing time) implies higher energy consumption. Conversely, the correlation is positive when using CHOCOLA: i.e., less processing time implies also less energy consumption.

The study shows that it is worth developing advanced transactional models as they not only help programmers simplify the implementation of such models but also bring time performance along with energy savings. This means that efforts to speed up the processing of transactions pay off in terms of energy.

Keywords: concurrency, energy consumption, benchmarking,
multi-threading, transactional model, STAMP

1. Introduction

Concurrent computing is nowadays omnipresent as most chips are multicore. Such chips can be found not only in state-of-the-art desktop computers (e.g. Intel's, AMD's, and Apple's M series), but also in mobile devices (e.g. Qualcomm's and Apple A series), IoT devices (e.g. Raspberry Pi), and in HPC systems (e.g. DOJO [1]). However, the full usage of such hardware depends on how the software that runs on top of it has been designed and implemented. The field of computer science aimed at making a contribution to deal with concurrent computing is known as concurrency, also named parallel computing, concurrent programming, and multiprogramming [2].

Contributions brought by the field of concurrency can today be found in many programming languages such as Clojure [3], Java [4] C++ [5], and Ocaml [6]. Such programming languages offer programmers a set of primitives [1] that make it possible to solve complex problems (like mutual exclusion) trivially.

A programmer uses these primitives to implement a particular concurrency model meant to introduce parallelism, and at the same time guarantee that certain properties hold. Examples of concurrency models that programmers may want to implement are locks, semaphores [7], futures, actors, and

---

[1] Some people prefer to talk about constructs.

transactions [8]. Examples of properties meant to be guaranteed are atomicity, consistency, isolation, and durability (known as the ACID properties [9]. In a nutshell, the implementation of a concurrent model brings speed (thanks to the parallel execution of processes) while preserving properties of interest. Therefore, it can be seen as a valid resource to enhance the time performance of the software.

It is believed that performant software is also energy performant. This fact a priory may sound reasonable as reducing the time to execute a piece of software may also reduce the energy it consumes during its execution. However, this is not totally accurate, as reducing the execution time of software (by parallelising tasks) increases the usage of the chip (more muti-cores are now used). This increase in the chip's usage results in more energy consumption.

Therefore, while extensive analysis of the time performance of concurrency models has been reported, the energy performance is (to the best of our knowledge) a big unknown.

In this paper, we aim to understand the energy footprint that comes along with concurrency models. We focus on a particular advanced concurrency model known as CHOCOLA [8], which combines three different concurrency models: actors, futures, and transactions. It is worth mentioning that in this study the transactional dimension provided by CHOCOLA is taken as main-class citizen to analyse both time and energy performance. That is the reason why we refer to it as an *advanced* transactional model (despite it also supports actors and futures). This built-in combination of multiple models makes it (to the best of our knowledge) the most advanced concurrency model available today. It is this characteristic that makes it an excellent candidate to find out energy footprint of transactional models. Since it is already known that it outperforms state-of-the-art transactional models when looking at time, we do want to know how it behaves regarding energy performance.

In this paper, we strive to 1) analyse the energy performance of an advanced transactional model (i.e. CHOCOLA) and a state-of-the-art model to determine if performance optimisation also brings energy savings, 2) determine how operating the transactional models under conditions that overpass the chip's maximal processing capacity impact on time and energy performance, and 3) find out whether time and energy performance are correlated.

The state-of-the-art transactional model used to compare CHOCOLA's energy performance is the same as the one used when assessing its time performance. This state-of-the-art model corresponds to the Software Transactional Memory (STM) provided by the STAMP [10] benchmark. Therefore, the assessment is done using the same well-known benchmarks and tools as done in the original study where CHOCOLA was reported. This allows any comparison to be fair. The results of this study have been put together according to the benchmarking guidelines indicated in the Empirical Standards for Software Engineering Research [11].

The main contributions of this paper are:

- confirming the successful replication of the original study reported in [8],
- providing a detailed description of the energy measurement steps taken to assess the transactional models,
- assessing the energy performance of both advanced and state-of-the-art transactional models to determine the impact that time performance enhancements have over energy performance,
- revealing the time and energy impact of using the transactional models under conditions that overpass the hardware's limits where it executes,
- confirming that time and energy are correlated, and that they not necessary need to be traded-off, and
- providing a replication package that allows the community to replicate the experiment to confirm the claimed findings.

The rest of the paper is organised as follows: Section 2 provides a brief introduction of the transactional models used in this study, as well as the measurement approach and tools used to cope with the observation of energy consumption. Section 3 explicitly states the goal of this study along with the research questions it aims to answer. The same section also explains the chosen means used to answer each research question. The organisation of the experiments performed in this study is explained in Section 4, whereas their actual executions are reported in Section 5. The experiment's results are presented in Section 6. These results are then used to revisit each research question and provide an answer for it. This is presented in Section 7. Section 8 presents works related. The limitations of this study are presented in Section 9. The paper closes with conclusions and future work (Section 10).

2. Background

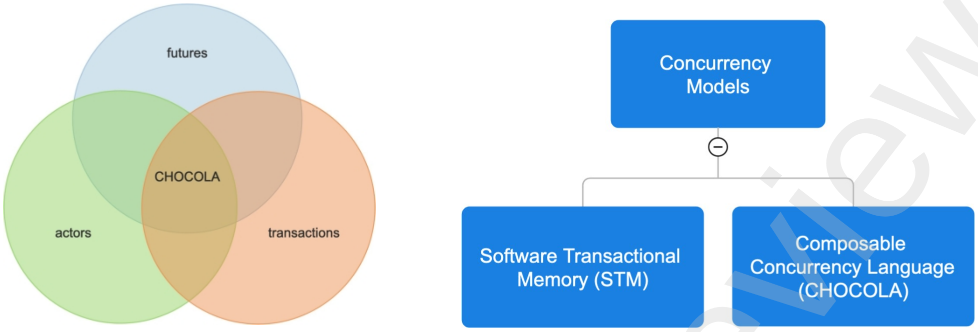
2.1. Concurrency and Energy

As CPUs plateaued in their clocking speeds, manufacturers focused more on putting more cores on CPUs instead of enhancing the speed of the cores [12]. This made concurrency (as a means to attain real parallelism) a valid approach for developers to implement more performant (i.e. faster) software. Thus, by using concurrency, the goal is to spread tasks amongst multiple cores such that they complete in a shorter period of time as if they were executed sequentially. However, when given extra thought, the benefits brought by concurrency to speed up the execution of a particular piece of software may be at the expense of other important quality as it is energy performance.

Energy E, measured in Joules (J), is the total dissipation of Power P, measured in Watts (W), over a given Time T, often measured in seconds (s), shown in the formula: $E = P * T$ [13]. For example; to finish a job in 10 seconds using 10 W, we would need to exert 100 J. To complete the same task with the same energy (i.e. 100 J), but in 5 seconds, we would have to increase the power to 20 W to compensate for the shorter time duration. Carrying this formula further, one might consider concurrency in software and apply the same logic: i.e., to be able to complete the same task in a shorter period, the software should require more power. In concurrency, this is done through *parallelism*. Parallelism is the utilisation of multiple threads in parallel to complete the same task more quickly. However, in doing so we are using more of the total capacity of the CPU, thus logically requiring more power. Looking back at the previous example, to go from 10 seconds to 5 seconds, we must use two times as more of the CPU as we did before, naturally needing more energy to be consumed.

2.2. CHOCOLA

Programming languages often allow the developer to mix different concurrency models during development freely. However, the mixed usage of concurrency models does not always guarantee the safety of their individual properties and can cause conflicts [8]. CHOCOLA is a programming language built as an extension of Clojure [3] that combines three concurrency models (futures, transactions, and actors), covering the three categories that concurrency models often fall into (deterministic, shared-memory, and massage-parsing), whilst guaranteeing all of their respective properties. Figure 1a



(a) (b)

Figure 1: Visual Representation on concurrency models and CHOCOLA

gives a visual representation of the concurrency models CHOCOLA is taking advantage of.

CHOCOLA is a resource meant to be used by programmers to deal with concurrency at the implementation level. The fact that it combines three different concurrent models allows programmers to have a large range of alternatives to satisfy requirements related to concurrency at the coding level. To the best of our knowledge, no similar work has been done up to now, so it makes CHOCOLA the most advanced development to implement concurrency models. In this paper, CHOCOLA will be referred to as an *advanced* transactional model due to the focus on transactions as a concurrency model while allowing futures and actors to be used in aid.

2.3. STAMP

STAMP [10] is a state-of-the-art benchmark suite exclusively built for Software Transactional Memory (STM), or transactional models, where a wide range of applications have been implemented with configuration options. The benchmarking of CHOCOLA is done via the *vacation2* benchmark, converted into Clojure from STAMP's *vacation* benchmark which is implemented in C. Vacation2 is implemented in Clojure and introduces the possibility to utilise actors and futures to compare STAMP's transactional model with their own advanced transactional model. Vacation2 is an application that simulates a real-life problem of planning a vacation by booking a hotel, a flight, and a car.

In this study we compare the two models (i.e. STM and CHOCOLA) via vacation2 as reported in the original study [8] and using exactly the same assets as provided in its replication package [14]. Figure 1b provides a visual representation of how STM and CHOCOLA relate to each other.

Is is worth recalling that similarly as done in the original study, four different variants of CHOCOLA are used during the benchmarking process. Each variant corresponds to a configuration based on the number of secondary actors available when processing transactions. Thus, each benchmarked model has an unique reference name to ease its identification: *STM, CHO-1, CHO-2, CHO-8*, and *CHO-64*, where STM stands for the state-of-the-art transactional model, CHO stands for CHOCOLA, and the numbers represent the number of secondary actors utilised by CHOCOLA.

2.4. RAPL

Intel's Running Average Power Limit (RAPL) [15] tool is an interface with which it is possible to monitor a system's various hardware and their power consumption. Research shows, the advantage of RAPL is that it offers easy and portable energy measurements with significant accuracy and little performance overhead, and is also *accessible via software*, no need for external hardware to get these readings, making RAPL a very promising tool [16]. RAPL allows the user to monitor different parts of different hardware with varying levels of isolation. For example; it allows for the monitoring of [16]:

- **power plane 0**: all processor cores on the CPU
- **power plane 1**: integrated GPU on the CPU
- **DRAM**: RAM attached to the integrated memory controller
- **package**: the entire CPU

For this research, we measure the *package* domain, as it is the most appropriate domain based on the compatibility constraints brought by the hardware where the experiments were executed. Package observes the entire CPU with no granular settings. It differs from **power plane 0** due to the fact that it does not only observe the cores, but any other component embedded into the CPU.

The interaction with RAPL is achieved via `perf` [17]. Perf is an monitoring open source software available for linux distributions. More explanation regarding `perf` can be found in Section 5.

3. Experiment Definition

3.1. Research Goal

By utilising the formulation proposed by Basili et al. [18], the goal of this study is to *analyse* the advanced transactional model CHOCOLA *for the purpose of* evaluation *with respect to* its energy consumption *from the point of view of* developers *in the context of* programming concurrent tasks.

3.2. Research Questions

Our study considers the following research questions.

- **RQ1:** *What is the energy performance of STM and CHOCOLA?* From the original study's findings it is already known that CHOCOLA has much better time performance than STM. However, we would like to know if this is also the case when looking at the energy dimension. The answer to this research question is searched using descriptive statistics.
- **RQ2:** *What are the effects over both time and energy when exceeding the maximum available computing capacity bound by the CPU?* To answer this question, we replicate the original study under the exact same conditions (except for hardware). As our hardware has a lower computing capacity, the replication allows us to observe a state space previously not observed by the original study. The significance of this research question is made clearer in the later sections. The answer to this research question is also searched using descriptive statistics.
- **RQ3:** *Is there a correlation between time and energy for each of the analysed concurrency models?* Finally, we want to determine (based on the collected data) if there exists a relationship between time performance and energy performance. The answer to this research question is searched using hypothesis testing.

3.3. Descriptive Statistics & Experimental Hypothesis

To be able to answer RQ1 and RQ2, we use descriptive statistics [19] (pp. 10-17) by presenting tables of the experiment's raw data, plotting line charts visualising the raw data, and using grouped bar charts [19] (pp. 63) to dive deeper into the data and analyse it. After presenting our data, we discuss the implications of our findings.

To be able to answer RQ3, we use hypothesis testing. The formulated *Null* and *Alternative* hypotheses are:

- $H1_0$: There is no correlation between energy consumption and time performance in the observed concurrency models.
  $H1_a$: There is a correlation between energy consumption and time performance in the observed concurrency models.

$$H1_0 : \rho(E, T) = 0$$
$$H1_a : \rho(E, T) \neq 0$$

$\rho(x, y)$ represents the correlation function. This function will be either Pearson or Spearman [19] (pp. 141). The exact function to be used to test the hypothesis depends on the distribution of the data.

4. Experiment Planning

4.1. Subject Selection

The research focuses on assessing a cutting-edge transactional model for which an implementation and experimental replication package already exists. The name of this transactional model is CHOCOLA [8]. The selection of this transactional model lies in the fact that it is the only model that combines 3 different concurrency models guaranteeing the properties of each of these models are preserved.

The second subject of this research is the STAMP implementation [10] of STM. Similarly as done in the orginial study where CHOCOLA was introduced, STM is taken as reference to assess both time and energy performance.

4.2. Experimental Variables

The dependent variables for all RQs are the energy consumption and time performance, measured in

4.3. Data Analysis

We analyse the experimental data collected *via* 4 main phases in this given order; normality testing, experiment replication validation, data exploration, and hypothesis testing.

Normality Testing: To get an insight into the normality of our data we use the *Shapiro-Wilk* [20] normality test, getting a statistical understanding of the data distribution. Depending on the results of these tests, we apply the appropriate statistical analysis methods to understand the meaning of our data.

Experiment Replication Validation: An introductory step for evaluating the quality of the collected data and understanding if the experiment has been replicated correctly. This is done by plotting charts displaying the median time performance of each transactional model to get a visual understanding of replication quality. If our data is normally distributed, we then compute the Pearson [19] (pp. 141) correlation coefficient with our data and their respective counterpart from the original study [8] to measure linear correlation. If our data is non-normally distributed, the Spearman rank-order coefficient [19] (pp. 141) is computed to do the same.

Data Exploration: We follow the replication validation step by presenting a summary of our data, visualising the measured energy and time performance data via various visual representations, and presenting tables to display descriptive statistical information meant to help answering RQ1 and RQ2.

Hypothesis Testing: In this last phase we answer RQ3 through hypothesis testing. We calculate the appropriate correlation coefficient as per the distribution of the data to measure the linear correlation between energy consumption and time performance across all pair-wise combinations of thread counts and secondary actors.

5. Experiment Execution

In our experiment, we replicate the experiment conducted in the original research using the replicationg package made available by CHOCOLA's authors [14].

It is worth indicating that the configuration parameters meant to set the initial conditions of the vacation2 case study remain identical as those reported in the original study. Having said that, it is important recalling that both STM and CHOCOLA are exercised by executing a version of vacation2 meant to attend 1000 booking requests [2]. As the load of each execution remains the same, this study does not include any analysis related to *memory* energy performance.

---

[2] A request represent the booking of a flight, hotel, and car for a family made of 5 people.

CPU	Intel i9-7900X @ 3.30GHz 20 cores [3]
Memory	8x 16 GB DDR4
OS	Debian 11, 64-bit

Table 1: Hardware Specs of Host System 'M1'

Differences between the original study and ours lay only at the level of the hardware used to execute the experiments. The specs of the host machine (**M1**) are shown in Table 1. This machine is used to observe not only the time performance (as done in the original study), but also the energy consumption of every execution of the experiment.

The experiment is conducted in the same way as the original experiment, with the `perf` [17] software measuring the energy consumption of each pair-wise combination of our independent variables. This process of measuring individual runs within the experiment was automated with a Bash script to log the results in between runs.

Each run is done by using `lein` [21], the same as in the original experiment. The amount of threads and secondary actors to execute the vacation2 case study are passed as parameters. Upon execution of `lein`, a new process is started. The amount of time and energy consumed by this process is measured using `perf`. The results of `perf` are logged, and a new run is started with the next combination of threads and secondary actors. A 10-second cooldown phase is implemented to account for internal validity concerns regarding *History*, as proposed as a best practice by Z. Ournani et al. [22] History validity refers to the change in time and setting in between the experiments ran on the same subject. [23] (pp. 106) In our case, we are trying to negate noise regarding CPU heating.

For this research, there is only one independent variable that we observe: the amount of *threads utilised*. There are 4 different configurations of CHOCOLA with varying secondary actor utilisation. Each pair-wise combination of thread and secondary actor configuration is run 30 times. This sample size is the same as in the original study [8]. This setup allows us to collect enough data for all pair-wise combinations, covering the real-life applications of multi-threading to the fullest capacity of our hardware.

---

[3] Reached by enabling hyperthreading over 10 physical cores.

RQ2 is formulated due to the fact that we explore a state space previously unobserved by the original study. Our host machine M1 has only 20 thr eads, while the machine used in the original study had 64. For the sake of consistency, we do not touch the replication package at all and build our experiment on top of the already prepared stack. Using `perf` on top of the replication package allows us to ensure there is not any noise introduced by us. However, the big processing capacity difference between our hardware and the one used in the original study provokes us to investigate this state space more, in terms of time, energy, and the trade-off between them. It is for this purpose that a second machine referred to as **M2** is used to collect extra evidence aimed at confirming initial findings obtained when using M1. This means that M2 is used to confirm whether the energy consumption trend of each concurrency models remains aligned with respect to the hardware's maximal available processing capacity. The specs of M2 are shown in Table 2.

CPU	Intel i7-4790 @ 3.60GHz 8 cores [4]
Memory	16 GB DDR3
OS	Ubuntu 20.04, 64-bit

Table 2: Hardware Specs of Host System 'M2'

As part of our efforts to (1) adhere to open science and (2) ease the reproducibility of the experiments reported in this study, a replication package has been prepared and made available to the community [24].

6. Results

6.1. Normality Testing

Before we do any analysis, we conduct normality tests [20] to understand the distribution of our data. In the original study [8], it was presented that the data was non-normally distributed. We want to conduct our own tests to not only to confirm the claims stated in the original study, but also to assess the data regarding energy consumption. As it is explained in Section

---

[4] Reached by enabling hyperthreading over 8 physical cores.

4.3, we first apply the Shapiro-Wilk normality test to assess if our data is normally distributed. The tests are conducted with sub-datasets containing time and energy performance data for each transactional model and each thread. This results in a total of 165 sub-datasets [5] for which normality tests need to be performed for time performance purposes. The same number of tests need also to be done for energy purposes. Each sub-dataset contains 30 values, which corresponds to the number of time the same experiment (i.e. execution of vacation2 with a same number of threads) is executed.

The results of these tests are heterogeneous. For STM, out of the 33 tests [6], 57% of them have a normal distribution of time performance, and 66% of them have a normal distribution of energy consumption. Similar trends are obtained when doing running the tests for each CHOCOLA configuration. Table 3 summarises the obtained results.

Transactional Model	Time	Energy
STM	57%	66%
CHO-1	78%	72%
CHO-2	27%	1%
CHO-8	60%	51%
CHO-64	78%	75%

Table 3: Percentages of Normally Distributed Data

As there is not a consistent pattern for normality, we are not able to decisively agree on the normality of our data distribution, neither for time nor energy. Thus, we conclude that our data is non-normally distributed and proceed to apply non-parametric statistical analysis methods. Recall that in the original study authors also concluded that data were not normally distributed.

6.2. Experiment Replication Validation

As discussed earlier, after running our experiments we conduct a replication validity analysis to ensure that the data collected through our measurements are in line with the original study's.

---

[5] There are 33 different amounts of thread, and 5 different transactional models - STM and 4 CHOCOLA configurations.

[6] A test for each thread, which varies from 1 up to 64 in steps of 2.

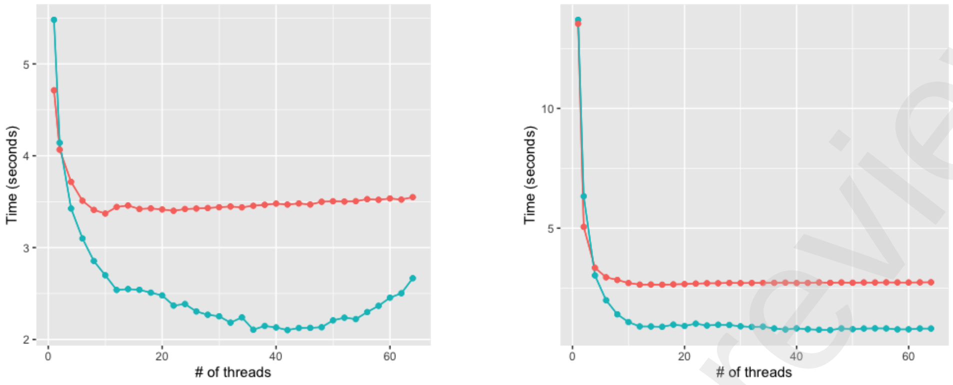
This replication validity analysis is conveyed by comparing the original study's data with ours. This comparison is done visually by plotting the median time performance that takes each transactional model to execute the case study (i.e. vacation2) for a given number of threads. It is worth recalling that the number of threads ranges from 1 until 64, jumping in steps of 2.

These plots are shown in Figures 2a, 2b, 2c, 2d, and 2e. On each figure is possible to visualise these median values from both the original study and our replication.

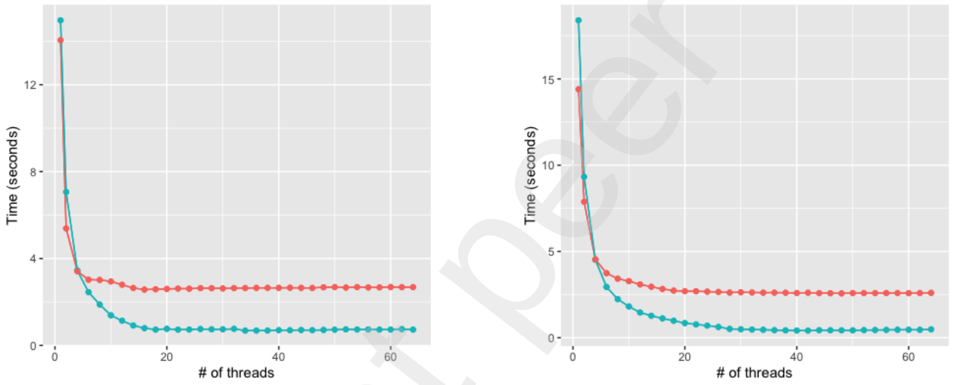
It can be seen that for each transactional model, the two data sets are similar in the sense that both respect the same trend. The fact that both median time performance values are not exactly is due to hardware differences, as discussed earlier. Beside the gap between lines due to hardware differences, the median values corresponding to the STM transaction model seem not to be strongly aligned. To gather more concrete evidence, liner correlation tests between the original study's data and ours for each transactional model is performed.

As we conclude the data is non-normally distributed the linear correlation between the two data sets is done with the Spearman rank-order correlation coefficient([19], pp. 141).

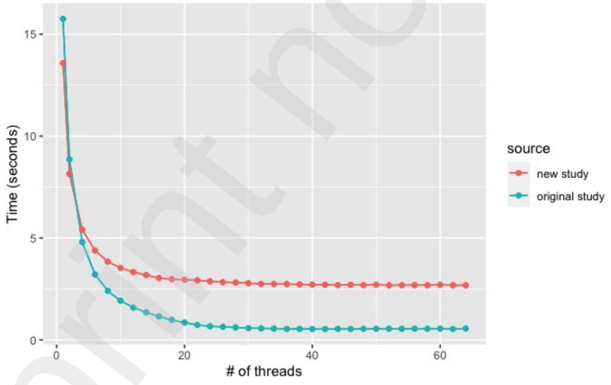
Figure 3a displays the correlation coefficients corresponding to each transactional model. As the coefficient ranges between 1 (very strong positive correlation) and -1 (very strong negative correlation), the figure allows the reader to quickly detect heterogeneity between the coefficients. While correlation coefficients for transactional models CHO-8 and CHO-64 seem to be very homogeneous with values higher than 0.75, CHO-2 shows a weaker but still valid correlation (in the range [0.25,0.50]). Conversely, coefficients for STM and CHO-1 are not only very weak, but also contradictory. Values shown on the left side of Table 4 provide the exact coefficient values (**r.value**) obtained from each correlation test for each model. These coefficient values along with the significance of each test (**p.value** would indicate that we did not successfully replicate the experiment. However, the hardware differences between both studies, allowed us to observe a state space not observed by the original study. This state space corresponds to the behaviour of the models when relying on a number of threads that go beyond the maximal processing capacity of the hardware. Recall that the original study's hardware had 64 cores, whereas ours 20. The analysis related to the performance of each transactional model when operating *above* the hardware's max processing capacity is done in Section 6.3.



(a) STM (b) CHO-1

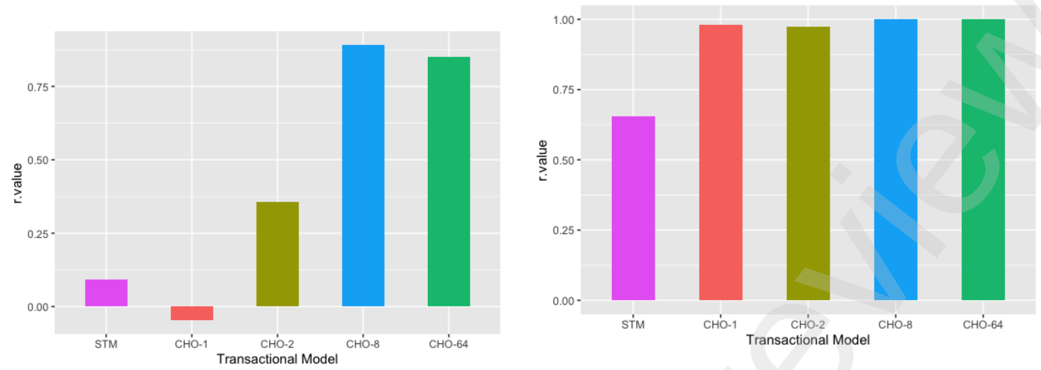


(c) CHO-2 (d) CHO-8



(e) CHO-64

Figure 2: Median time performance results from the original study (in green) and our new study (in red).



(a) Over total number of threads.

(b) Up to hardware (M1)'s maximal processing capacity.

Figure 3: Strength and direction of the relationship between the original's study data and ours using Spearman's rank correlation coefficient

model	r.value	p.value
STM	0.09	0.6
CHO-1	-0.04	0.79
CHO-2	0.35	**0.04**
CHO-8	0.89	**3.59e-08**
CHO-64	0.85	**2.23e-07**

model	r.value	p.value
STM	0.65	**0.03383**
CHO-1	0.98	**2.2e-16**
CHO-2	0.97	**2.2e-16**
CHO-8	1	**2.2e-16**
CHO-64	1	**2.2e-16**

Table 4: Experiment replication correlation for total number of threads **(left)** and up to hardware (M1)'s maximal processing capacity **(right)** data

Based on this fact, for the correlation analysis to be fair, it should be carried out on data that meet similar conditions with respect to the hardware: i.e., the number of threads on which the transactional model operates does not exceed the hardware's max processing capacity. Therefore, we run the correlation test for each model again, but including data only up to 20 threads.

Figure 3b displays the correlation coefficients corresponding to each transactional model for these new tests. They look now very homogeneous. Their exact values along with the significance of each test are shown on the right side of Table 4. For all benchmarked transactional models we have now a very strong homogeneous correlation between our results and the results of the original study.

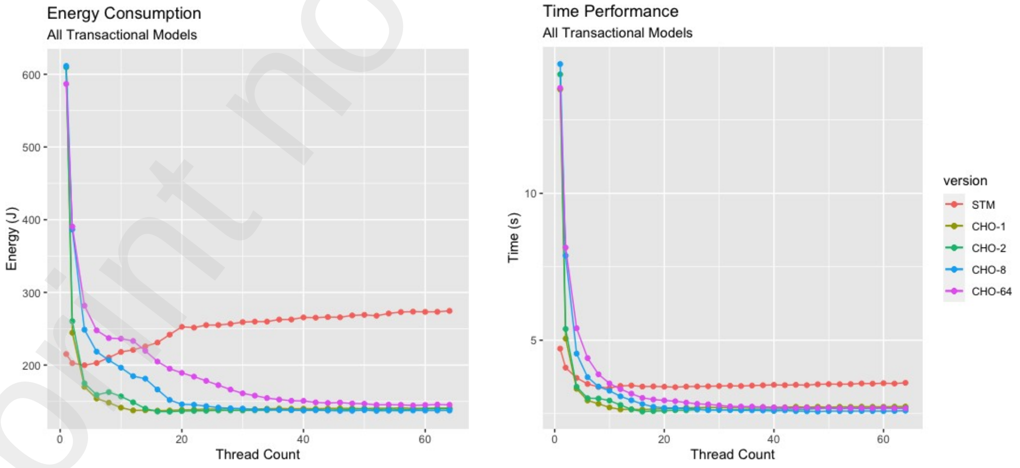
In conclusion, we believe that we present sufficient evidence proving that we were successful in replicating the original study [8], thus the collected data should be valid to allow us to answer the stated research questions.

6.3. Data Exploration

In this section, we plot charts and display tables aimed at easing the understanding of the data through the execution of the experiment using the different transactional models under analysis. These plots and tables are used in Section 7 as supportive evidence to answer the stated research questions.

It is worth recalling that the experiment consists of executing the vacation2 case study using one particular transactional model. Whereas the transactional model remains fix all along the execution of the experiment, the number of threads (independent variable) it operates on vary from 1 up to 64 in steps of 2. The experiment ends once the execution has been completed over all threads. The experiment is run 30 times for each transactional model. The collected data have proven not to be normally distributed, thus the median of 30 collected points for each observation is used as reference to summarise the behaviour of each transactional model over the observed range of threads.

Figures 4a and 4b show the median energy and time performances, respectively, of each measured transactional model across each thread count tested.



(a) Energy

(b) Time

Figure 4: M1 Total Energy & Time Performance

These plots show that CHOCOLA has high values both for time and energy performance when the number of available threads is lower than 4. That means that the lack of threads affects the time and energy performance of CHOCOLA very negatively. The situation changes when the number of available threads on which CHOCOLA can operate is increased. Therefore, these charts give hints regarding how CHOCOLA is expected to behave when deployed on a hardware that either does not have a lot of cores available (worse case) or does have a large number of cores (better case). The same plots can be used to understand the performance of STM when operating over different settings. While the energy gets worse when increasing the number of threads available for its execution, the time performance does the same but at a lower rate. Therefore, these plots suggest that CHOCOLA has better time and energy performance than STM.

In order to get a more precise view of the collected data, we use descriptive statistics to get a more analytical view of the data tendency. Table 5 displays statistical summaries of STM's and each of the CHOCOLA configurations' energy and time consumption.

Tx. Model		**Min.**	**1st Qu.**	**Median**	**3rd Qu.**	**Max.**
STM	***E (J)***	199.8	231.0	259.8	267.9	274.6
	T (s)	3.371	3.431	3.469	3.510	4.713
CHO-1	***E (J)***	137.3	139.4	140.2	140.8	609.9
	T (s)	2.638	2.705	2.718	2.732	13.537
CHO-2	***E (J)***	135.9	137.7	138.7	140.0	609.8
	T (s)	2.570	2.638	2.656	2.685	14.049
CHO-8	***E (J)***	137.0	137.5	139.0	166.4	611.5
	T (s)	2.570	2.589	2.617	2.826	14.4
CHO-64	***E (J)***	144.5	147.2	158.0	204.8	586.8
	T (s)	2.679	2.700	2.745	3.035	13.586

Table 5: Energy (E) and time (T) consumption per transactional (Tx.) model.

Based on these statistical summaries, *CHO-2* is the most efficient transactional model as it has the best (lowest) median (2.656 secs) and minimum (2.570 secs) time performance. In terms of energy consumption, again, *CHO-2* has the most efficient results with the lowest median (138.7 J) and minimum (135.9 J). However, all CHOCOLA configurations have shown a close performance among each other, both for time and energy performance. What the data show actually is that CHOCOLA has outperformed STM both in energy consumption and time performance efficiency in every aspect.

Furthermore, to explore the difference between our hardware and the original study's hardware, we split the data into smaller sub-datasets. In the original study, authors relied on a hardware that had 4 CPUs, each with 16 cores, which gave a total of 64 cores (that may explain why they scale threads up to that value when running the experiments). We rely on M1 to replicate the original study, which has 20 cores. As explained earlier, we do not want to change the settings of the original study despite of using a hardware with less processing capacity. Having replicated the original study under the same conditions, but on a different hardware allows us to observe the behaviour of each transactional model when operating below and above the maximal processing capacity of the hardware. In our case this threshold corresponds to 20 threads as M1 counts 20 cores.

Figure 5 shows the speed-up values for each transactional model. For coverage, the data is split into two subsets, below the max capacity and above the max capacity. The formula used to calculate the speed value S_i is:

$S_i = \frac{V_1}{V_i}$ where;

V_1: the energy (or time) when threat count is equal to 1,

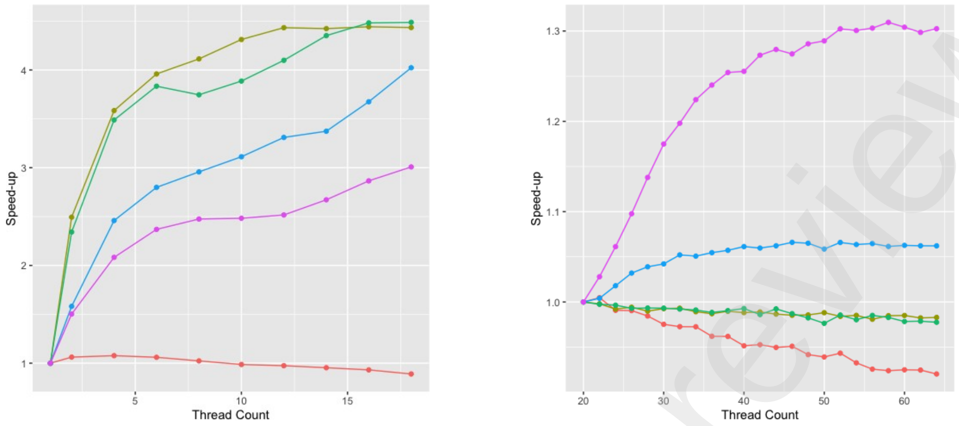
V_i: the energy (or time) value when threat count is equal to i, with

i ranging from 2 to 64 in steps of 2.

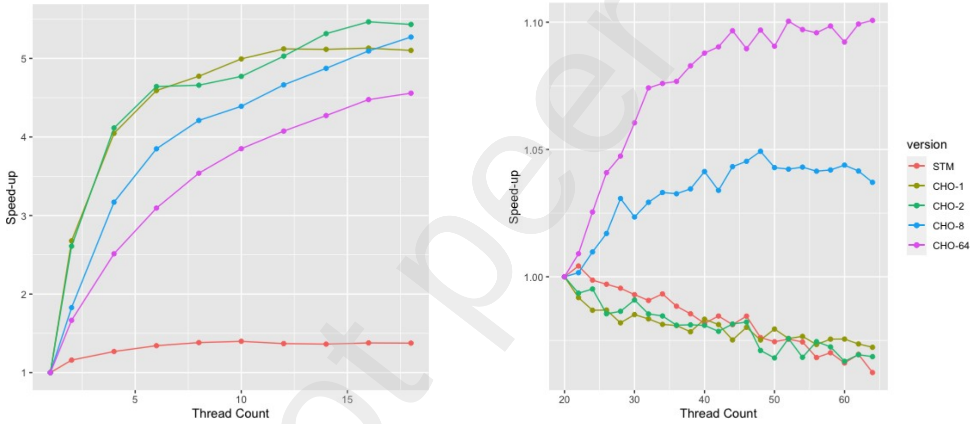
Figures 5a and 5b show the effect of the number of threads that are utilised by each transactional model on *energy consumption*.

Figure 5a shows the energy speed up of each transactional model when remaining below the maximal processing capacity, whereas Figure 5b shows what happens when operating the models beyond the max capacity.

We notice that STM does not show a decrease in energy consumption as we increase the degree of concurrency neither below nor above the max capacity. While CHO-1 and CHO-2 slightly join STM's trend when operating above the max capacity, we see a significant change in energy consumption going from a single thread to the max capacity of M1. Furthermore, CHO-8 and especially CHO-64 further show the power of the CHOCOLA model by lowering energy consumption even more. While configuring the transactional



(a) Energy Below (b) Energy Above



(c) Time Below (d) Time Above

Figure 5: Speed-up for energy and time (more is better i.e. less time/energy consumption)

model to operate with 64 threads (which are not actually there) we see a speed-up of 1.3 in energy consumption compared to the max capacity of the system. The degree with which this decrease is observed seems to be correlated with the number of secondary actors utilised by CHOCOLA; as 1 and 2 secondary actors do not show this behaviour, 8 shows it but not as much as 64 which shows a significant performance boost.

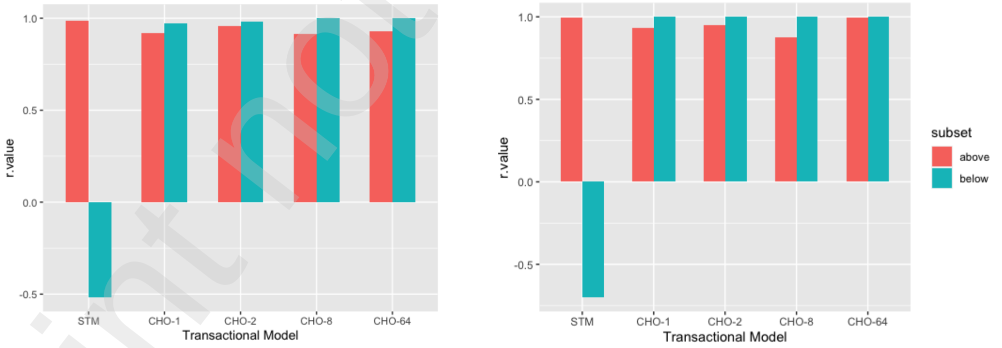
Figures 5c and 5d show the effect of the number of threads that are utilised by each transactional models on *time consumption*. The former figure shows when operating each transactional model below M1's max capacity, whereas the latest when doing so above M1'a max capacity. We see that for all transactional models, there is a significant performance boost for time when operating below the maximum capacity. When operating above the max capacity, CHO-8 and CHO-64 again show a performance boost in time, where using 64 secondary actors yields a speed-up of 1.1 in time consumption. A significant finding was that CHO-2 outperformed all other models in both energy and time performance while operating below the max capacity with speed-ups of 4.49 and 5.43, respectively.

Based on these figures, CHO-2 with 18 threads is the best configuration out of all tested models. This is due to the fact that there are 20 cores available in M1, and 18 threads and 2 secondary actors appear to be the best trade-off between threads and secondary actors for the models we test.

6.4. Hypothesis Testing

After doing the necessary normality tests, validating that the experiment was successfully replicated, and exploring the data collected, we test the hypothesis derived from RQ3.

As we concluded that the data is non-normally distributed, Spearman rank-order correlation coefficient is used for all concurrency models to calculate the linear correlation between time performance and energy consumption. We also split up the transactional models below and above the max capacity.



(a) M1

(b) M2

Figure 6: Strength and direction of the relationship between energy and time performance for each transactional model both below and above the hardware's max processing capacity.

Figure 6a and Table 6 provide information about the correlation between time and energy, both above and below M1's processing capacity threshold for each transactional model.

model	subset	r.value	p.value
STM	below	-0.52	0.1
	above	0.98	**1.8e-06**
CHO-1	below	0.97	**2.2e-16**
	above	0.92	**3.3e-06**
CHO-2	below	0.98	**2.2e-16**
	above	0.96	**2.5e-06**
CHO-8	below	1	**2.2e-16**
	above	0.91	**8.9e-10**
CHO-64	below	1	**2.2e-16**
	above	0.93	**3.2e-06**

model	subset	r.value	p.value
STM	below	-0.7	0.2333
	above	0.99	**2.2e-16**
CHO-1	below	1	**0.0166**
	above	0.93	**1.6e-07**
CHO-2	below	1	**0.0166**
	above	0.95	**5.3e-08**
CHO-8	below	1	**0.0166**
	above	0.87	**7.1e-07**
CHO-64	below	1	**0.0166**
	above	1	**2.2e-16**

Table 6: Energy and Time correlation for each transactional model as observed on **M1** (left) and **M2** (right).

Despite the figure and the table provide the same information, they complement to each other. The figure allows the reader to visually determine not only how strong is the correlation between the two concerned variables, but also whether the correlation is positive or negative. The table provides the analytical values that allow the reader to determine the exact same information, plus to see the significance of each statistical test.

The data is aligned with Figure 5, confirming that there is a correlation between time performance and energy consumption (**RQ3**). All CHOCOLA models have very strong positive correlations. However, for STM we see that while operating below the max processing capacity, the time performance and energy consumption are negatively correlated. On the other hand, just like the CHOCOLA models, while operating above the processing capacity threshold, time and energy are strongly positively correlated.

We see that in Table 6 for STM while operating below the maximum capacity our correlation test results are not significant enough (> 0.05) for both M1 and M2. This is because neither M1 nor M2 has enough cores relative to our sample size (up to 64).

In the case of M2, the *p.value* is too high, meaning that our *r.value* of -0.7 is not significant. However, although we see the same problem with M1 too, the *significance is greater*. M2 has 8 cores, meaning that when we are assessing correlation *below the max capacity* we are looking at the data points for 1, 2, 4, 6, and 8 threads utilised, giving us five data points, while most of our sample size is assessed in the *above the max capacity* category. M1 on the other hand has 20 cores, giving a total of 11 data points. Normally, we would not be able to reject the null hypothesis under these conditions. However, we believe that a trend is visible where M1 has more data points under max capacity, and has greater significance. Thus if these tests were run on a machine with a much greater number of cores (ideally 64), we would be able to reach confident significance levels for STM below the max processing capacity too.

Combining all of our observations, we can understand the energy-time correlation, and also the effects of concurrency on them:

- **STM:** we see that time and energy are negatively correlated when operating *below* the max processing capacity. On Figure 6a this can be visualised by the bar going under zero for STM. Looking at Figures 5a and 5c we can observe the trend, while energy consumption is *increasing* time consumption is *decreasing*. However, while operating *above* the max capacity they are both showing an upwards trend, thus they are positively correlated. This can be visualise by the bar that almost reaches 1 (0.98 to be exact) on the left side of Figure 6a. Thus, this visual manner of presenting the correlation between time and energy below and above the max processing capacity is meant to ease the understanding that both dimensions under certain conditions may be traded off (e.g. when operating below max processing capacity), whereas under different conditions one may become a byproduct of the other.

- **CHO-1 and CHO-2:** Energy and time are strongly positively correlated both below and above the maximum capacity. When looking at the trends (see Figures 5a and 5c), we see that there are significant incentives below the maximum capacity; however, while operating above there are minor penalties.

- **CHO-8 and CHO-64:** Energy and time are again strongly positively correlated both below and above the maximum capacity. Trends below the maximum capacity are similar to the other two CHOCOLA configurations. However, unlike the other ones, the trends are different above the maximum capacity. When operating *above* the maximum capacity (meaning that the model is trying to use cores that are not physically available) we see a decrease in time and energy consumption. While it is weaker for CHO-8, for CHO-64 we observe a decrease of energy consumption and time execution.

Figure 6b and the right side on Table 6 present the results (both visual and analytic) of the correlation between time and energy based on data collected using M2 machine. The fact of having reached such high correlation values consistently across two different machines provide extra validity to our findings.

In conclusion, we can confidently reject the null hypothesis. This means that there is a significant correlation between time and energy in concurrency systems.

7. Discussion

In this section, we revisit our initial research questions and discuss the implications of our findings for developers and users of transactional models. **RQ1:** *What is the energy performance of STM and CHOCOLA*? It is already known that CHOCOLA outperforms the state-of-the-art STM implementation in terms of time. This was shown in the original study where CHOCOLA is presented [8]. Our findings show that CHOCOLA does the same with energy performance as well, providing a significant energy consumption decrease. Figure 5 shows that CHOCOLA benefits from great energy consumption as well as time performance trends, and can cut costs on both dimensions significantly. This is promising evidence for developers to prefer advanced transactional models like CHOCOLA to ensure the lossless integration of different concurrency models. In conclusion, not only does CHOCOLA reduce time costs but also energy costs. Therefore, it is very much worthwhile to continue investing in the development of advanced transactional models such as CHOCOLA. This conclusion then represents a point in favour for those who believe that energy performance is a byproduct of time performance [25].

RQ2: *What are the effects of exceeding the maximum available computing capacity bound by the CPU over both time and energy?* During our reproduction phase we noticed that we were exploring a state space previously not considered by the original study. This new state space represent the usage of the concurrency models under conditions that exceed the processing capacity of the hardware hosting the model. When trying to utilise more threads than there are available, according to our findings (see Figures 5b and 5d) STM consumes more energy and takes longer to finish its task. However, CHOCOLA, when configured with at least 8 secondary actors manages to squeeze out more efficiency from the CPU; and when set with 64 secondary actors managed to save almost 25% extra energy while also being almost 10% faster, even though there were only 20 cores available in the host.

We believe the driving force for the success of CHOCOLA especially when operating over the hardware's max processing capacity comes from the combination of transactions and actors concurrency models (and distinctive characteristic of CHOCOLA). Conversely, when operating STM below the max processing capacity, there already are conflicts at the level of transactions. When operating it above the max capacity, on top of the conflicts at the transaction level, we introduce conflicts at the hardware level as now more threads are trying to utilise the same amount of cores. In the case of CHOCOLA, the introduction of actors makes the management of transactional conflicts so much more efficient that, if configured correctly, CHOCOLA can benefit from this at the hardware level too and reach energy efficiency levels STM can not.

Similar results are found when using M2. Despite it holds a different processing capacity threshold, the transactional models have near identical behaviour as when operated over M1 (see Figures 6a and 6b). This is significant evidence that investing in advanced transactional models such as CHOCOLA is very beneficial both from a time and energy performance perspective. A priory, this finding seems again to be supporting those researchers who argue that energy savings are merely a byproduct of time performance. However, when focusing on STM we can see that energy and time are traded off, specially when such state-of-the-art transactional model is operated without exceeding the hardware's processing capacity. Thus, there are cases when improvements in one dimension may lead to losses on the other one.

RQ3: *Is there a correlation between time and energy for each of the analysed concurrency models?* We observe a strong correlation between time performance and energy consumption in both STM and CHOCOLA. According to our findings (see Figures 6a, 6b, and Table 6), CHOCOLA shows a very strong positive correlation between time performance and energy consumption across the board. Conversely, STM shows a negative correlation when being operated without going beyoind the hardware's max processing capacity. The trends of STM changes when exceeding the hardware's processing capacity. Ultimately, this means that developers should prioritise the usage of such *advanced* transactional models that are more energy-friendly when building large concurrent systems.

8. Related Work

CHOCOLA is a resource meant to be used by programmers to deal with concurrency at the implementation level. The fact that it combines three different concurrency models (i.e. futures, transactions, and actors) allows programmers to have a large range of alternatives to satisfy requirements related to concurrency at the coding level. To the best of our knowledge, no similar work has been made up to know, making it the most advanced composable concurrency model available to the community. This made it an excellent candidate to analysis its energy performance as it is already known it outperforms state-of-the-art transactional models in term of time performance. Despite (to the best of our knowledge) the fact that nobody has performed a similar study, we are aware of certain works that report analysis of energy consumption. Following, we indicate these works, indicating similarities and differences.

The work of Mishra et al. [26] presents a new inter-process communication (IPC) algorithm aimed at reducing the locking of the CPU. Reducing the locking of the CPU results in allowing it to perform other tasks without wasting its energy on useless spinning. Whereas IPC mechanisms are central to the management of multiple threads running on multi-core chips, such as mechanisms at the level of operative systems, this is very different from CHOCOLA's: built-in programming languages primitives aimed to easily solve the implementation of actors, futures and transactions concurrency models (either individually, or composite).

Yet another study that focuses on energy performance is the work of Z. Ournani et al. [27] where the factors of study are different implementations of Java Virtual Machines. Despite the common interest in energy performance and the use of benchmarking as a research methodology to conclude, the difference between this study with ours is on the asset under evaluation.

The work of Pereira et al. [28] gets closer in terms of abstraction level to our work. They focus the analysis of energy consumption on state-of-the-art programming languages. That means that this work targets the main kind of first-class citizen like us: programmers. However, while it is worth knowing the energy performance of a particular programming language, the programming language Clojure (the one used to implement CHOCOLA's model) has not been included. Moreover, none of the problems used as benchmarking when evaluating the programming languages is similar to the *Vacation2* application.

The work of Magalhães et al. [29] falls into the same category as Pereira's: i.e. the analysis of programming languages. However, in this case, they focus their analysis of energy performance on the multi-threading aspects of the programming languages. This study relies on the NAS Parallel Benchmark [30] to perform the study, which has been conceived to evaluate the performance of parallel computers and related tools for high-performance computing. Thus, whereas the number of commonalities between this study and ours is large, the difference still remains on the asset of assessment: programming languages vs. implementation of advanced transactional models.

Last but not least, it is worth mentioning that all the related works reported previously have relied on RAPL [15] as the mechanism to measure energy consumption, similarly as we have done.

9. Threats to Validity

In this section, we discuss the threats to the validity of our study based on the categorisation proposed by Campbell and Cook [31]

9.1. Internal Validity

Before the experiment, all background processes that could be terminated were terminated, ensuring no extra load was put on the system, and a cooldown phase was introduced throughout the experiment to negate *History* concerns, as per the instructions of Z. Ournani et al. [22]. To remediate for *Selection Bias*, we extensively tested for all possible parameters, thus there was no selection process.

9.2. External Validity

A possible threat to external validity could be in *Ecological Validity*, residing in hardware specifics. The data used to answer the research questions brought by this study was collected using a specific host hardware (see Table 1 for its specs). For example, we observed the performance of the transactional models *with respect to* our hardware specs, such as the total available cores. Although replications of our experiments on different hardware could yield different performance data, we believe that the behaviour of the transactional models *with respect to* the available cores in the host hardware should remain unchanged. We have conducted experiments on a second host (see Table 2) to compare the performance of the transactional models *with respect to* the available cores in the host. The results of this comparison are able to back up our claims.

In terms of *Interaction of Setting and Treatment*, there is a threat to validity in Section 6.4 due to our low *p.values*. However, we believe we have shown enough evidence that it is not a crucial problem that is limiting our capabilities, or misguiding our conclusions. To accommodate for this problem completely, the experiment should be replicated with a machine that has a greater number of cores (ideally 64).

9.3. Construct Validity

To accommodate for *Measurement Error*, we paid attention to the benchmarking tool used to measure energy consumption. As discussed in Section 5, `perf` was the chosen measurement tool to observe the energy consumption of each experiment execution. This tool is made available as a package for linux distributions. Beside being open source, this package is maintained and updated frequently. Additionally, `perf` relies on RAPL [15], which is proven to be reliable and accurate regarding the estimation of software energy usage. We took a careful approach to create the most efficient environment for `perf` by eliminating all external noise we could. Thus, we believe that we have dealt with validity concerns in this domain well.

9.4. Conclusion Validity

In terms of *Low Statistical Power* concerns, almost all of our results in our applied statistical tests yielded very strong and conclusive results, with the exception of STM operating under hardware's max processing capacity (see Section 6.4). We believe that we have shown enough evidence proving that this is not a problem as the trend shows that: if the test were replicated on a machine with a greater number of cores, then it would allow the sample size *under max capacity* to be increased; so we would have correlation test results with much better significance (i.e. p-value much lower than 0.05) .

We believe we worked with a sufficient sample size, enough to be sure about the odds of our results being by chance. Furthermore, we were careful with the environment setup and experiment execution. Close attention was paid to the design of dependent and independent variables of the research, and all external noise was dealt with to the best of our capabilities. We believe our results are valid and reliable.

To conclude, all concerns mentioned in this section will be treated with care in any future work that follows to further enhance the quality of this research and ultimately raise awareness about energy consumption through true and reliable data and scientific work.

10. Conclusion and future work

In conclusion, our study shows that it is worth developing performant advanced transactional models such as CHOCOLA as it not only helps programmers to simplify the concerns brought by concurrency, but also reduces energy consumption while maintaining the already achieved enhancement in time performance. The collected evidence has shown that performance and energy consumption are correlated when using these concurrency models to deal with scenarios like the one brought by the STAMP's vacation benchmark. This evidence has shown that the correlation is negative when relying on on the standard STM model, meaning that higher performance (i.e. less processing time) implies higher energy consumption. Conversely, when using CHOCOLA, the evidence has shown that the correlation is positive: i.e., less processing time implies less energy consumption. We believe that our presented results agree with the idea that energy is a byproduct of time performance, and that time performance optimisation is a valid method of optimising energy performance, see Section 1.

Currently, the energy performance analysis of CHOCOLA focuses only on the processing consumption (i.e. the energy consumed by the processor), without considering memory usage. Thus, as part of our future work, is planned to consider memory usage in our measurements: i.e. the same kind of experiments reported in this work will be done again but include a new independent variable meant to vary the "data load" of the Vacation2 application. Also, STM experiments will be replicated to accommodate for *Interaction of Setting and Treatment* (see Section 9.2) and *Low Statistical Power* (see Section 9.4) validity concerns.

References

[1] E. Talpes, D. D. Sarma, D. Williams, S. Arora, T. Kunjan, B. Floering, A. Jalote, C. Hsiong, C. Poorna, V. Samant, J. Sicilia, A. K. Nivarti, R. Ramachandran, T. Fischer, B. Herzberg, B. McGee, G. Venkataramanan, P. Banon, The Microarchitecture of DOJO, Tesla's Exa-Scale Computer, IEEE Micro 43 (3) (2023) 31–39. `doi:10.1109/MM.2023.3258906`.

[2] L. Lamport, An incomplete history of concurrency chapter 1. 1965–1977, in: Proceedings of the 2014 ACM Symposium on Principles of Distributed Computing, PODC '14, Association for Computing Machinery, New York, NY, USA, 2014, p. 291. `doi:10.1145/2611462.2611514`.

[3] R. Hickey, The clojure programming language, in: Proceedings of the 2008 symposium on Dynamic languages, ACM, 2008, p. 1.

[4] J. Gosling, B. Joy, G. L. Steele, G. Bracha, A. Buckley, The Java Language Specification, Java SE 8 Edition, 1st Edition, Addison-Wesley Professional, 2014.

[5] N. M. Josuttis, The C++ Standard Library: A Tutorial and Reference, 2nd Edition, Addison-Wesley Professional, 2012.

[6] X. Leroy, D. Doligez, A. Frisch, J. Garrigue, D. Rêmy, J. Vouillon, The ocaml system: Documentation and user's manual, INRIA 3 42.

[7] F. B. Schneider, On Concurrent Programming, Springer-Verlag, Berlin, Heidelberg, 1997.

[8] J. Swalens, J. D. Koster, W. D. Meuter, Chocola: Composable concurrency language, ACM Trans. Program. Lang. Syst. 42 (4) (jan 2021). `doi:10.1145/3427201`.

[9] J. Gray, A. Reuter, Transaction Processing: Concepts and Techniques, Morgan Kaufmann, 1993.

[10] C. C. Minh, J. Chung, C. Kozyrakis, K. Olukotun, Stamp: Stanford transactional applications for multi-processing, in: Proceedings of the IEEE International Symposium on Workload Characterization (IISWC), 2008, pp. 35–46. `doi:10.1109/IISWC.2008.4636089`.

[11] P. Ralph, Acm sigsoft empirical standards released, SIGSOFT Softw. Eng. Notes 46 (1) (2021) 19. `doi:10.1145/3437479.3437483`.

[12] D. Patterson, The trouble with multi-core, IEEE Spectrum 47 (7) (2010) 28–32, 53. `doi:10.1109/MSPEC.2010.5491011`.

[13] G. Pinto, F. Castor, Energy efficiency: A new concern for application software developers, Commun. ACM 60 (12) (2017) 68–75. `doi:10.`

1145/3154384.
URL https://doi.org/10.1145/3154384

[14] J. Swalens, Original study's reproduction package.
URL https://github.com/jswalens/vacation2

[15] E. Rotem, A. Naveh, A. Ananthakrishnan, E. Weissmann, D. Rajwan, Power-management architecture of the intel microarchitecture code-named sandy bridge, IEEE Micro 32 (2) (2012) 20–27. doi:10.1109/MM.2012.12.

[16] K. N. Khan, M. Hirki, T. Niemi, J. K. Nurminen, Z. Ou, Rapl in action: Experiences in using rapl for power measurements, ACM Trans. Model. Perform. Eval. Comput. Syst. 3 (2) (mar 2018). doi:10.1145/3177754.

[17] Perf wiki.
URL https://perf.wiki.kernel.org/index.php/Main_Page

[18] V. R. Basili, G. Caldiera, H. D. Rombach, The goal question metric approach, 1994.
URL https://api.semanticscholar.org/CorpusID:13884048

[19] R. Pagano, Understanding Statistics in the Behavioral Sciences, Cengage Learning, 2012.

[20] J. Faraway, Extending the Linear Model with R: Generalized Linear, Mixed Effects and Nonparametric Regression Models, Chapman & Hall/CRC Texts in Statistical Science, CRC Press, 2016.
URL https://books.google.de/books?id=ODcRsWpGji4C

[21] Leiningen home page.
URL https://leiningen.org/

[22] Z. Ournani, M. C. Belgaid, R. Rouvoy, P. Rust, J. Penhoat, L. Seinturier, Taming energy consumption variations in systems benchmarking, in: Proceedings of the ACM/SPEC International Conference on Performance Engineering, ICPE '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 36–47. doi:10.1145/3358960.3379142.
URL https://doi.org/10.1145/3358960.3379142

[23] C. Wohlin, P. Runeson, M. Hst, M. C. Ohlsson, B. Regnell, A. Wessln, Experimentation in Software Engineering, Springer Publishing Company, Incorporated, 2012.

[24] Anonymous, Energy Performance Analysis of Advanced Transactional Models (Sep. 2023). doi:10.5281/zenodo.8341561.
URL https://doi.org/10.5281/zenodo.8341561

[25] D. Mahajan, C. Blakeney, Z. Zong, Improving the energy efficiency of relational and nosql databases via query optimizations, Sustainable Computing: Informatics and Systems 22 (2019) 120–133. doi:https://doi.org/10.1016/j.suscom.2019.01.017.
URL https://www.sciencedirect.com/science/article/pii/S2210537918301112

[26] R. Mishra, I. Ahmad, A. Sharma, An energy-efficient queuing mechanism for latency reduction in multi-threading, Sustainable Computing: Informatics and Systems 30 (2021) 100462. doi:https://doi.org/10.1016/j.suscom.2020.100462.
URL https://www.sciencedirect.com/science/article/pii/S2210537920301864

[27] Z. Ournani, M. C. Belgaid, R. Rouvoy, P. Rust, J. Penhoat, Evaluating the impact of java virtual machines on energy consumption, in: Proceedings of the 15th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM), ESEM '21, Association for Computing Machinery, New York, NY, USA, 2021. doi:10.1145/3475716.3475774.

[28] R. Pereira, M. Couto, F. Ribeiro, R. Rua, J. Cunha, J. a. P. Fernandes, J. a. Saraiva, Energy efficiency across programming languages: How do energy, time, and memory relate?, in: Proceedings of the 10th ACM SIGPLAN International Conference on Software Language Engineering, SLE 2017, Association for Computing Machinery, New York, NY, USA, 2017, p. 256–267. doi:10.1145/3136014.3136031.

[29] G. G. Magalhães, A. L. Sartor, A. F. Lorenzon, P. O. A. Navaux, A. C. Schneider Beck, How programming languages and paradigms affect performance and energy in multithreaded applications, in: 2016 VI Brazil-

ian Symposium on Computing Systems Engineering (SBESC), 2016, pp. 71–78. `doi:10.1109/SBESC.2016.019`.

[30] J. Löff, D. Griebler, G. Mencagli, G. Araujo, M. Torquati, M. Danelutto, L. G. Fernandes, The nas parallel benchmarks for evaluating c++ parallel programming frameworks on shared-memory architectures, Future Generation Computer Systems 125 (2021) 743–757. `doi:https://doi.org/10.1016/j.future.2021.07.021`.
URL `https://www.sciencedirect.com/science/article/pii/S0167739X21002831`

[31] T. Cook, D. Campbell, Quasi-Experimentation: Design and Analysis Issues for Field Settings, Houghton Mifflin, 1979.