Dynamic Energy Modelling for SoC Boards: Initial Experiments

Adam Seewald, Emad Ebeid, Ulrik Pagh Schultz
SDU UAS Center, University of Southern Denmark, Odense, Denmark
[ads|esme|ups]@mmmi.sdu.dk

***Abstract*—Embedded Systems are key components of any robotic system and are heavily involved in most capabilities ranging from basic control to complex object detection techniques. Whereas low-end embedded systems typically are very predictable in terms of power consumption, this is normally not the case for high-end, heterogeneous embedded systems incorporating CPU and GPU cores. In this paper, we present initial work with an experimental setup aimed to be used for measuring the power consumption of different heterogeneous embedded systems using benchmarks representing realistic workloads. Our approach sets the stage for future work concerning energy models capable of predicting power consumption for high-end, heterogeneous embedded systems.**

***Index Terms*—System-on-a-chip (SoC), Energy Modelling, GPU, CPU**

I. Introduction

Mobile robots and factory process controllers are all examples of Embedded Systems [1]. Although they have spread widely to everyday objects, they remain a crucial component of advanced systems such as aerial vehicle controllers. State-of-the-art controllers are able to perform real-time autonomous operations and have become extensively involved in various tasks that otherwise required a human operator's interaction. A major challenge today is the development of autonomously operating agents able to perform tasks independently. To this end, visual sensing techniques have been integrated into the control pipeline of robotics systems such as Unmanned Aerial Vehicles (UAVs), in order to enhance their navigation and guidance skills [2]. These techniques are widely used to perform autonomous tasks, including path planning, obstacle avoidance, environment mapping and object recognition.

With this work, we propose a measurement approach that will allow gathering overall power consumption data for heterogeneous embedded systems. The data will be used in a future method for prediction of the power usage of different component configurations within a full Aerial Robotics application on a UAV use case. An effective power predicting model can create a significant impact on the UAV community, by increasing flight operational time and by allowing real-time computations. In the past, such operations were often carried out on other devices, by a ground operator or statically prior to the beginning of the mission. The approach that we intend to pursue in the final instance also applies to a more general set of embedded devices, such as IoTs, wearable electronics, and critical control systems, where energy efficiency is relevant in order to achieve specific functionalities.

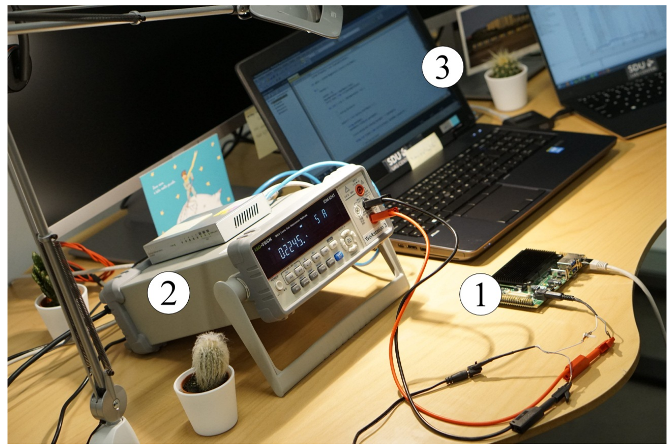

Fig. 1. A photo showing the experimental setup's implementation

II. Background

Execution time and security properties of software are extensively discussed in the available literature, but very little treatment has been reserved to the energy efficiency, essentially due to its unpredictability. It still remains one of the most important challenges in future computer technologies: its importance spans from miniature embedded sensors to wearable electronics, from the individual computer to the data centers [3].

Calore presents an approach for measuring the power efficiency for High Performance Computing (HPC) systems [4]. An external board was used to measure the power consumption but the data was collected only from the NVIDIA Tegra K1 device on one specific benchmark. We aim to base our work on a similar setup.

Approaches to minimize UAV power consumption, such as the work done by Kreciglowa, aims to determine the best trajectory generation method for an aerial vehicle to travel from one configuration to another [5]. Uragun suggests the use of power efficient components [6]: an energy efficient UAV system can be built using both emerging technologies, as conceptual product development, and energy efficient components. Kanellakis et al. affirm that integrating visual sensors in the UAV ecosystem still lacks solid experimental evaluation [2]. They suggest that in order to save energy, the available payload for sensing and computing has to be restricted.

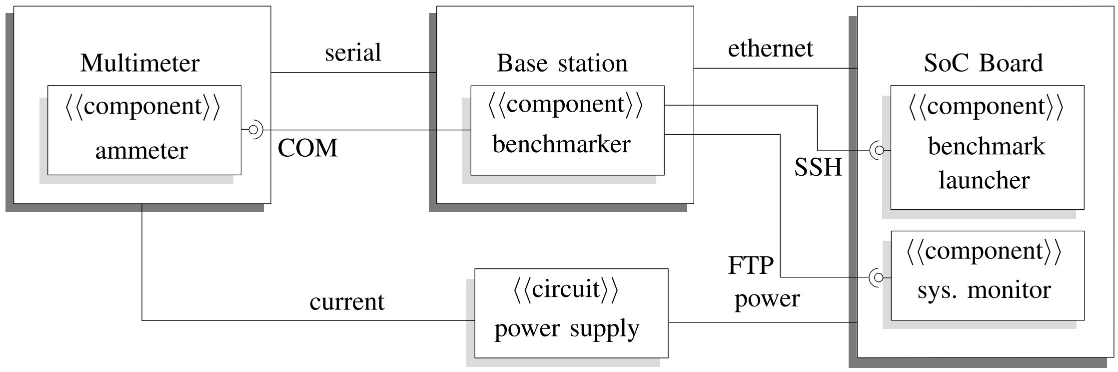

Fig. 2. UML diagram of the experimental setup

III. Methods

A. Hardware Experimental Setup

In order to measure an embedded board's overall power consumption, a setup consisting of three nodes was built, as shown in the UML diagram of Figure 2. The first node is the board under analysis. The setup permits to use any embedded board that runs a Linux OS with the SSH protocol port enabled. The second node is the multimeter that performs the sampling of the power consumption at a specific time intervals that can be configured. The sampling frequency was set to 5 Hz due to the instrument's limits, in our case the Good Will Instrument GDM-8341 digital multimeter. The multimeter is connected to the board using a serial connection from the power supply's plus phase. The third node is the master base station. On one side, the station is interfaced to the system using standard TCP/IP protocols: the SSH protocol to control the board, and the FTP protocol to retrieve the data. On the other side, the multimeter is connected to the station using the serial connection.

ARM CPUs and NVIDIA GPUs are among the most energy efficient hardware [7]. Recently both architectures have been combined in the NVIDIA Tegra systems-on-chip. All our experiments were done on an NVIDIA Tegra K1 board that incorporates ARM CPU and NVIDIA GPU cores. Figure 1 shows the experimental setup's implementation and it is (labels show the node number).

B. Software Experimental Setup

An automated setup to perform the measurement and to evaluate the outcomes was implemented. The base station, the third node of the general power measurement setup, stores a list of benchmark available on the board that is specified before the analysis. Various combinations of the benchmarks can additionally be included in the list in order to measure the behavior of a fully loaded system.

In detail, the base station starts to log power consumption sampled from the circuit and retrieved from the multimeter via serial connection back to the station. Next, the base station sends a request to start a subset of all the possible benchmarks that can be performed on a specific embedded board via SSH connection. The benchmark must be stored and compiled on the board prior to the experimental evaluation. The base station starts to evaluate power-to-compute E_C, expressed in Watt per hour, and time-to-compute T_C, expressed in seconds, as well. A logging process on the board is also started to store the data regard the CPU and GPU usage.

The log is retrieved via FTP connection when each of the desired benchmarks in the subset terminated its execution, after which a check of the data consistency is made on the base station. All the data and metrics are stored and are plotted using a MATLAB script. The base station checks the consistency of the data by comparing the data retrieved from the multimeter with the data retrieved from the board. For the time being for these initial experiments, the benchmarks were repeated until a stable result was found; an automated setup that ensures proper statistical validity is considered future work.

C. Benchmarks

In our simplified case, we use two different benchmark programs, along with their combinations. First, we use the Prefix-sum algorithm [8]. Prefix-sum is a well known apparently-only sequential algorithm that can be easily parallelized on both many-core architectures and GPUs. Its applications include parallel implementations of deleting marked elements from an array (stream-compaction), radix-sort, solving recurrence equations, solving tri-diagonal linear systems, and quick-sort [9]. The algorithm runs on both CPU and GPU.

Second, we use YOLO, a standard computer-vision algorithm based on the Darknet framework [10]. The algorithm is used to detect objects within one image on a pre-trained neural network. With this algorithm, the object detection is reframed as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities. The experimental setup tries to mimic a real UAV companion board application, used to both evaluate the environment through visual sensing, and to perform energy heavy computations.

D. Metrics

The time-to-solution metric indicates the total time expressed in seconds needed to perform all the benchmarks in the subset. The energy-to-solution indicates the total amount of energy expressed in Watt per hour needed. This metric is obtained using the expression Equation 1, where ν indicates the sampling frequency, I_i the sampled value of the current in Ampere, and V the system's total voltage potential expressed in Volts:

$$E_C = \frac{\sum_{i=1}^{T_C \cdot \nu} (I_i \cdot V)}{3.6 \cdot 10^3 \cdot \nu} \tag{1}$$

The total energy consumption in one hour is evaluated with the expression of Equation 2:

$$\bar{E}_C = \frac{E_C \cdot 3.6 \cdot 10^3}{T_C} \tag{2}$$

IV. Results

A summary of the average data obtained running a specific subset of the two benchmarks algorithms used can be seen in Table I. The behavior is almost as expected. With an increased complexity of the tested benchmark set, the power consumption increases considerably, almost redoubling its overall consumption.

	Benchmark	T_C [s]	E_C [Wh]	$\bar{E}_C$ [Wh]
B_1	⟨PrefixSum⟩	22.355	0.035584	5.73035
B_2	⟨Darknet⟩	81.360	0.134181	5.93719
B_3	⟨Darknet,PrefixSum⟩	57.602	0.130437	8.15208
B_4	⟨2×Darknet⟩	69.452	0.189735	9.83480
B_5	⟨3×Darknet⟩	87.819	0.234248	9.60266
B_6	⟨4×Darknet⟩	95.951	0.241116	9.04648
B_7	⟨5×Darknet⟩	108.04	0.280437	9.34419

TABLE I: Energy/power-to-solution measurements

The energy profile of benchmark B_1 that shows an algorithm running on both CPU and GPU cores and performing PrefixSum algorithm, can be observed in Figure 3. The CPU metric from the figure expresses the average CPU usage over the four cores. The energy profile of the benchmark B_3 that runs two algorithms, the Darknet running only on one CPU core and the PrefixSum on the GPU, is shown in Figure 4. Finally, the energy profile of benchmark B_5 with a fully saturated CPU is shown in Figure 5.

V. Discussion

An unexpected behavior can be observed for a single versus multi-core occupancy within the same application. This is probably due to the Dynamic Voltage and Frequency Scaling (DVFS) mechanism on the Tegra K1 board. We can observe that an increment in the frequency indicates an increment in the power consumption, i.e., the benchmarks sets B_3 and B_4 are significantly faster than B_2 but consume almost double overall energy per hour. The frequency can be adjusted accordingly in order to compute a solution faster and with more energy usage or to compute a slower solution and with an energy saving.

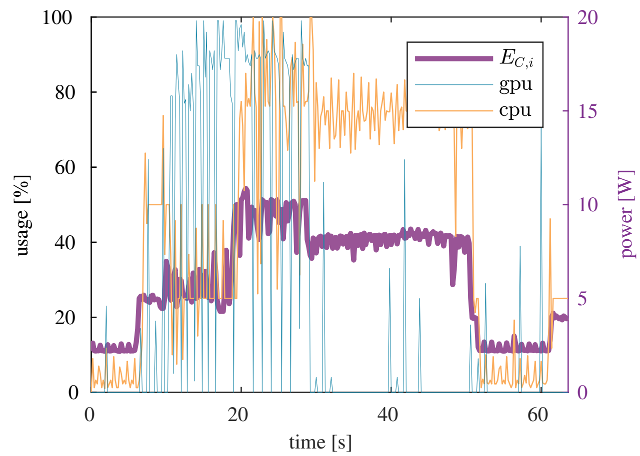

Fig. 3. Plot of the B_1⟨PrefixSum⟩ benchmark

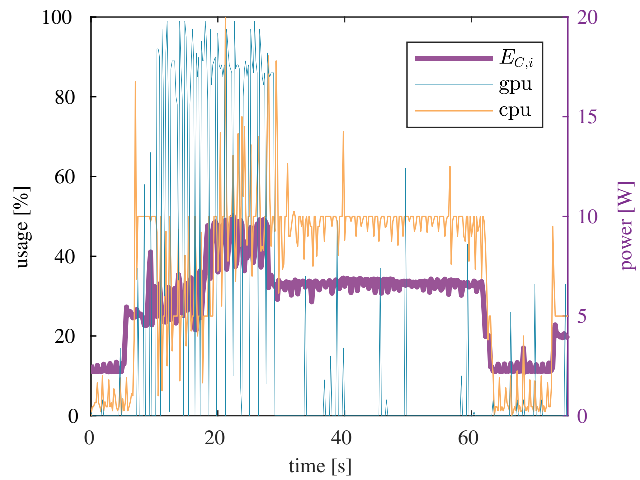

Fig. 4. Plot of the $B_3 = $ ⟨Darknet,PrefixSum⟩ benchmark

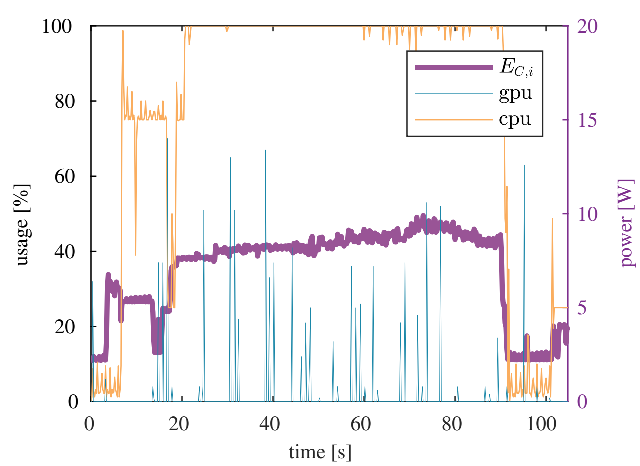

Fig. 5. Plot of the $B_5 = $ ⟨3×Darknet⟩ benchmark

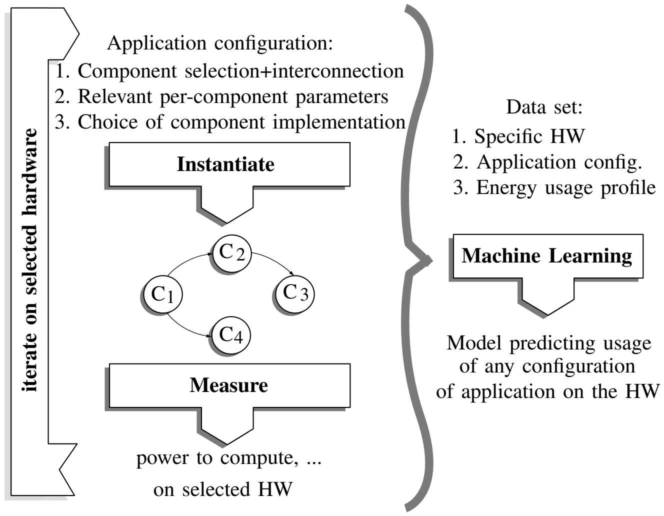

Fig. 6. Overview of the proposed approach to dynamic energy modeling

VI. Conclusion

In this work we described an experimental setup to measure the power consumption on System-on-Chip heterogeneous embedded systems. We developed a partially automated benchmark board, able to operate independently of both the system under analysis and the desired benchmark set. All experiments were made using a well-known embedded board, the NVIDIA Tegra K1, that has been extensively adopted for a wide variety of Aerial Robotics scenarios. As concrete benchmarks, we opted for two well-known algorithms that can be easily replicated and therefore applied to different scenarios as well. We observed the behavior of a benchmark running on the CPU, on the GPU, and both in combination. We thus set the stage for a future, deeper analysis to be used for a more complex and energy-oriented optimization framework.

VII. Future Work

Our immediate future work concerns establishing a set of benchmarks corresponding to a realistic UAS workload and evolving the proposed experimental setup to achieve a higher degree of accuracy, for example by increasing the sample rate and properly handling dynamic effects such as DVFS.

On a longer term, we are interested in using the proposed experimental setup to automatically derive an energy model capable of predicting power usage for high-end, heterogeneous embedded systems. We aim to do this by gathering data for a given board and for a specific application, and using this data to drive a machine learning process that from the application configuration can predict the power usage for this given board. This approach is illustrated in Figure 6. We assume a component-based approach to developing applications, as proposed for the TeamPlay project [11]. Here, a set of components can be composed in different ways and connected by dataflows. Each component has a set of configuration parameters and may have multiple implementations with different properties in terms of time, energy, and security (for example due to being implemented for CPU or for GPU execution). For a given concrete configuration (component graph, configuration, and implementation), we would then automatically measure the power consumption, and record this as data to be used for a subsequent machine learning process.

Acknowledgements

This work is supported and partly funded by the European Union's Horizon2020 research and innovation program under grant agreement No. 779882 (TeamPlay).

References

[1] L. P. Kaelbling, *Learning in embedded systems*. MIT press, 1993.

[2] C. Kanellakis and G. Nikolakopoulos, "Survey on computer vision for UAVs: Current developments and trends," *Journal of Intelligent & Robotic Systems*, vol. 87, no. 1, pp. 141–168, 2017.

[3] Y.-H. Lu, A. M. Kadin, A. C. Berg, T. M. Conte, E. P. DeBenedictis, R. Garg, G. Gingade, B. Hoang, Y. Huang, B. Li *et al.*, "Rebooting computing and low-power image recognition challenge," in *Computer-Aided Design (ICCAD), 2015 IEEE/ACM International Conference on*. IEEE, 2015, pp. 927–932.

[4] E. Calore, S. F. Schifano, and R. Tripiccione, "Energy-performance tradeoffs for HPC applications on low power processors," in *European Conference on Parallel Processing*. Springer, 2015, pp. 737–748.

[5] N. Kreciglowa, K. Karydis, and V. Kumar, "Energy efficiency of trajectory generation methods for stop-and-go aerial robot navigation," in *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*. IEEE, 2017, pp. 656–662.

[6] B. Uragun, "Energy efficiency for unmanned aerial vehicles," in *Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on*, vol. 2. IEEE, 2011, pp. 316–320.

[7] V. P. Nikolskiy, V. V. Stegailov, and V. S. Vecher, "Efficiency of the Tegra K1 and X1 systems-on-chip for classical molecular dynamics," in *High Performance Computing & Simulation (HPCS), 2016 International Conference on*. IEEE, 2016, pp. 682–689.

[8] W. D. Hillis and G. L. Steele Jr, "Data parallel algorithms," *Communications of the ACM*, vol. 29, no. 12, pp. 1170–1183, 1986.

[9] S. Sengupta, A. E. Lefohn, and J. D. Owens, "A work-efficient step-efficient prefix sum algorithm," in *Workshop on edge computing using new commodity architectures*, 2006, pp. 26–27.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[11] "Press release announcing start of TeamPlay," https://teamplay-h2020.eu/index.php?page=press-release, accessed: 2018-11-24.