# Self-aware Sensing and Attention-based Data Collection in Multi-Processor System-on-Chips

Nima TaheriNejad, M. Ali Shami and Sai Manoj P. D.
Institute of Computer Technology
TU Wien, Vienna, Austria
Email: {nima.taherinejad, muhammad.shami, sai.dinakarrao}@tuwien.ac.at

***Abstract*—Self-awareness is the foundation for many of the nowadays desired system characteristics, such as self-optimization and self-adaption. This awareness is rooted in observation and sensory data obtained by the system regarding itself and its environment. Given the important role which data collection plays in creating this awareness, we believe that it merits more attention than it has so far received. For example, increasing the amount of collected data can overload the system with increased computational cost, communication load, and power consumption. Self-awareness can help the system by making data collection smarter and better oriented. In this paper, we propose an attention-based data collection method, inspired by self-awareness, and exploit its potential in the context of Multi-Processor System-on-Chips (MPSoCs). Our case study shows that this method can reduce the computation and communication load related to processing sensory data up to 95%, at the cost of a negligible overhead at the sensor node.**

## I. Introduction

While the number of transistors continue to increase with every technology generation, since 2004, the increase of frequency has leveled off at around 4 GHz [1]. Ever since, parallelism and specialization have been the main agents of performance and efficiency improvements. Parallelism leads to multi- and many-core processors, and specialization manifests itself as heterogeneity and the increasing use of accelerator units for specific tasks such as video, audio, baseband, and graphics processing. As a consequence, resource management has become more and more complex.

Although available resources keep growing, available power envelop does not follow the same rate of growth, mostly due to temperature constraints. This leads to the peculiar situation, that all resources can never be active at the same time [2], a phenomenon that has been termed dark silicon [3]. The underlying reason is that the voltage cannot be scaled down sufficiently [4, Figure 13] to compensate for the increase of transistors on chip. Hence, resource management is becoming more complicated due to ever increasing tight constrains.

Fault management has several dimensions due to various causes of faults, masking effects and various available techniques that have to be combined to provide a reliable system behavior in the presence of adverse failure scenarios [5], [6]. Aging and wear-out phenomena are temperature and thus activity dependent [7]. In order to maximize lifetime, the load should be distributed such that resources which have undergone a higher aging rate in the past or have a higher temperature currently are spared from high loads .

In summary, on-chip resource management is a formidable challenge since it has to consider and balance workload and performance, real-time and safety constraints, power consumption and temperature, battery load and lifetime, fault and failure mitigation, aging and lifetime policies. All these objectives depend on variables, measured continuously during operation, such as temperature, power consumption, workload, and occurrence of faults. This continuous monitoring, itself comes at some overhead costs such as the area and power consumption of the sensors themselves, communication between sensors and the control unit, and the computation cost at the management unit. Self-awareness [8], however, holds the promise for a management level on top of local controllers that can handle incomplete or unreliable measurements, and reconcile different objectives for complex computing systems with many dynamically changing resources and using an optimized amount of resources. Another promise of self-awareness, on which we focus in this work, is its capability to reduce some of the overhead caused by continuous monitoring, without causing any loss of performance.

The rest of this paper is organized as the following: In Section II, we briefly review the concepts of self-awareness, based on which we propose a new attention-based technique in Section III. In Section IV, we present our simulation results and discussion before concluding the paper with Section V.

## II. Self-awareness, Sensing and Attention

As elaborated in detail in the literature, e.g., [9], [8], self-awareness implies abstraction of primary input data; a mapping into the semantic domain with respect to what is desirable and what is not, keeping track of the history, a model of goals and a model of the environment. Self-awareness plays a critical role in control loops such as Observe-Decide-Act [10], Observe-Orient-Decide-Act [11], or MAPE-K [12], in which it provides the foundation for effective decision making. In fact, it also includes a model of this decision making process. However, only recently the significance of the observation and abstraction process has received a more proper acknowledgement [13]. Between the measurement of data and decision making, the steps of data abstraction, their assessment in semantic, context sensitive terms, and their importance with respect to given goals and expectations are indispensable [13]. In addition, it is inefficient and often an obstacle to collect and process all possible data. In contrast, the higher level

978-1-5090-4991-2/17/$31.00 ©2017 IEEE

context and the measured data together should guide the data collection and processing, via a mechanism that has been termed “attention” [13], [14].

Several research groups have proposed solutions for on-chip resource management based on sensors and control loops. Zeppenfeld et al. [15] propose an autonomic SoC platform for dynamic resource management. The data abstraction and assessment is done via Learning Classifier tables which are based on rules and have a limited capability to be dynamically updated. SEEC [10] is a framework for performance and resource management. It monitors performance with application heartbeats (meaning that the platform checks the application performance in regular intervals, called heartbeats), and allocates resources accordingly to meet the registered objectives. In essence, the application itself is responsible for the data abstraction mechanism by providing the interpretation of the data in the application context. CPSoC [8] has one of the most elaborate mechanism for sensing and processing of data. A large set of sensors is distributed over the chip, collected via a dedicated network and processed by custom hardware called Introspective Sentient Units (ISUs) [8]. Elaborate processing and interpretation of the data is done and used for prediction of performance, power consumption, and other vital properties. It involves all layers, from the circuit to the architecture and middleware to the application, which allow decision making at the appropriate level [8].

These promising examples highlight that more work is required to fully exploit the the processes of sensing, abstraction and attention. For instance, as elaborated by TaheriNejad et al. [13], meta-information about collected data is as important as the data themselves, e.g., in order to assess the reliability of the data and give the measured data their deserved weights.

In the following sections we further elaborate on this aspect of self-awareness and show how the processing of temperature measurements can be reduced up to 95% with a context specific attention mechanism that filters out irrelevant data.

## III. Proposed Technique

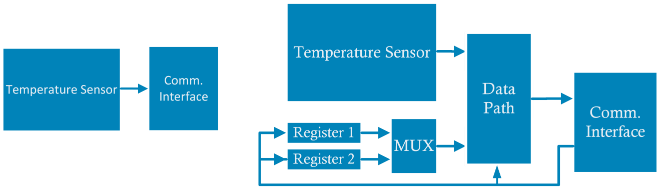
Nowadays, numerous sensors are added on die to help achieving a more comprehensive monitoring and a better control over various parameters on chip. With the exponential growth of the number of sensors on chip, e.g., on a many-core system with more than hundred cores, computational and communication resources needed for processing the sensory data in order to make an appropriate control decision is not anymore insignificant. To alleviate this problem, conventional methods such as down-sampling could be applied, for which the Nyquist rate is the limit to avoid loss of information. However, if we consider application and context we can considerably reduce the number of samples without missing critical events.

Taking temperature as an example, in a heterogeneous Multi-Processor System-on-Chip (MPSoC), multiple cores and various accelerators will hardly ever be loaded uniformly. Most applications will load some of the cores and accelerators heavily while they barely use some others. As a consequence, the temperature in the heavily used areas may reach the temperature limit which requires tight monitoring to keep it contained. Other areas, which are not critically heated up, on the other hand, do not need such frequent measurements. The reason being that critically high temperatures can negatively affect the performance and lifetime of the chip more than temperatures in the vicinity of the normal operation.

Hence, we propose a new attention-based sensor architecture where each sensor sends its observed value for processing, only when an important event has occurred (e.g., the temperature increases above a threshold). In this case the threshold reference is always updated after a temperature reading is sent to the control block. This distinguishes the proposed method from traditional thresholding techniques where the reference is an absolute value [16]. The proposed technique is also different from watchdog timer [17] since there is no timeout in this technique. Moreover, it does not store temperature values the way a buffer does to compensate for slow processing of data [18].

The proposed architecture, Fig. 1(b), needs a simple data path (which could be only an adder), a multiplexer, two registers, and a small control logic in addition to the traditional architecture, Fig. 1(a). Using this hardware, the mode of operation can still be flexibly adjusted by management units higher in the control hierarchy. For example, the registers could be filled with the average values observed by the temperature $\pm\Delta$, where $\Delta$ could be the standard deviation of the signal, statistically obtained and calculated by the higher level processing units. $\Delta$ could also be a percentage (decided by the designer) of the average value. In cases where certain absolute changes are of interest, $\Delta$ could be an absolute value too. Similar scenarios can be implemented, using the latest reported value instead of the average value. Furthermore, instead of symmetric thresholds ($\pm\Delta$), asymmetric thresholds could be used ($+\Delta_1$, and $-\Delta_2$).

The proposed strategy will slightly increase the computational hardware and processing effort at the sensor node, however, we contend that this overhead is relatively minor compared to the processes already running on digital sensors for calibration or the area used by analog sensors. This overhead is even more negligible if the sensor communicates to higher control units through methods such as a Network-on-Chip (NoC). The major impact of this approach is reducing the cost of processing data and decision making at the system



(a) Traditional

(b) Proposed

Fig. 1. Block Diagram of Temperature Sensors: (a) Traditional Design, (b) Proposed Attention-based Design.

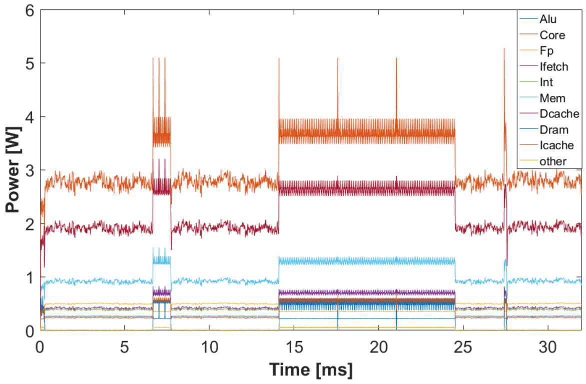


Fig. 2. Power profile of different units of the studied core.

level as well as unburdening the communication.

## IV. Simulation Results

### A. Simulation Set-up: Temperature Data

In order to show the efficacy of our proposed attention-based sensor architecture, we performed an experiment using an Intel Nehalem based 64-bit processor with an area of $32mm^2$, running SPLASH-2 benchmark suit. Barnes algorithm was simulated on a single core processor in Snipersim [19] and taken as basis. The resulting power profile was adapted to obtain an extended period of high power consumption (and consequently an extended variation in the temperature profile). Hotspot [20] is employed to obtain the temperature profile for the system. Figures 2 and 3 show the obtained power and temperature profiles, respectively.

### B. Experiments and Results

Since the focus of our experiments is on the number of samples that high-level controllers need to process for temperature management, it is only fair to assume that the original data (sampled at $16\mu s$) would be down-sampled based on the Nyquist rate. Therefore, we first extracted the frequency elements of the temperature data in MATLAB® using FFT analysis. Based on the Nyquist criteria, a down-sampling factor of 100 was applied which left us with 20 samples during the $32ms$ length of the experiment. We can see in Fig. 4, an example showing that the down-sampled signal (blue-dotted-cross) follows the original signal (black-solid-unmarked) very closely and without loss of any particular information.

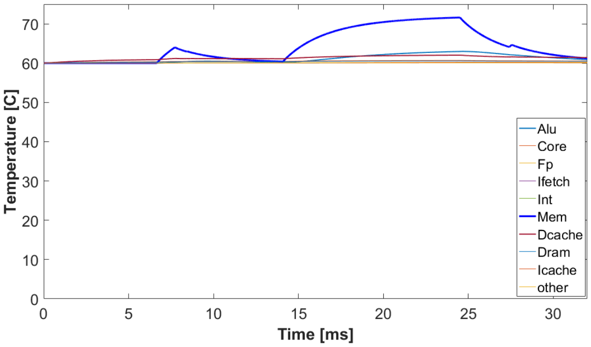


Fig. 3. Temperature profile of different units of the studied core.

TABLE I
Number of Attention-Based Samples ($\Delta$s are in $°C$)

| Unit | $\Delta = 1$ | Imp. | $\Delta = 2$ | Imp. | $\Delta = 5$ | Imp. |
|---|---|---|---|---|---|---|
| Memory | 13 | 35% | 9 | 55% | 4 | 80% |
| ALU | 4 | 80% | 2 | 90% | 1 | 95% |
| D-Cache | 2 | 90% | 2 | 90% | 1 | 95% |

To model the proposed architecture we loaded the registers with their latest reported value $\pm\Delta$, where $\Delta$ was an absolute temperature value, given that in MPSoC management schemes based on temperature, actual values and absolute changes are of most interest. We ran the experiments for three different values of $\Delta$, namely 1, 2 and 5°C. That is, if the temperature had a change beyond 1, 2 or 5°C, compared to its last reported value, the sensor will report the new value. Arguably, these $\Delta$s can provide the temperature management unit, with enough, if not overly fine resolution for management [21]. Our results show that for the majority of the 10 sensors (7 of them) in this study, attention-based sensors reported only one value (their initial value) as compared to 20 values which should have been reported after down-sampling. Number of the samples reported by sensors which reported more than one value are inserted in Table I. Next to each value stands the percentage of improvement, compared to the number of samples after Nyquist down-sampling. Lastly, Fig. 4 shows the curves of the original, Nyquist down-sampled, and attention-based sampled ($\Delta = 1°C$ and $5°C$) for the Memory unit of the processor.

### C. Discussion

As we could already see in Fig. 3, majority of the units monitored by the sensors, have a fairly monotonic, flat temperature profile. So it is not surprising to see that 7 out of 10 sensors report only one value in the beginning and no other values during the experiment. In other words, during this $32ms$

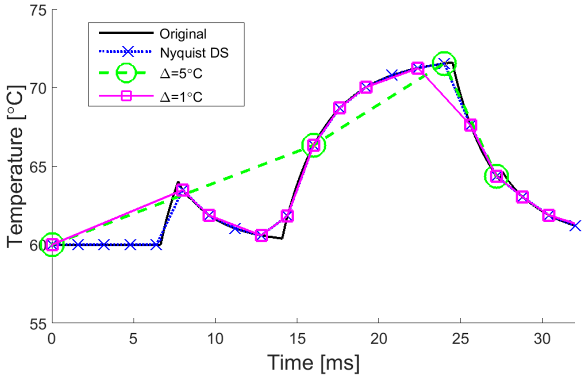


Fig. 4. Temperature profile for 'Memory' before (black-solid) and after Nyquist down-sampling (blue-dotted-cross), and attention-based reduced versions for $\Delta = 1°C$ (pink-solid-square) and $\Delta = 5°C$ (green-dashed-circle).

they experience less than one degree Celsius change in temperature, which can be simply disregarded by the temperature management unit. That means a 95% saving of the processing in the temperature management unit and communication load between the sensor and the management unit.

Regarding other signals, let us consider the worst case scenario, the unit with most changes in this experiment; Memory. By looking at its temperature profile in Fig. 4, we can see that the signal after down-sampling using proposed attention-based method, with $\Delta = 1°C$, follows the original signal very closely. That means that even with the (overly fine) resolution of $1°C$, a 35% saving on the number of samples which should be processed is achieved with virtually no loss of information. However, it can be argued that most temperature management units do not need to and do not react to $1°C$ change of temperature. We contend that $5°C$ is a fine-enough resolution for majority of temperature management tasks. In which case, as we can see in Fig. 4, the most important events (the rise, peak, and fall of temperature) in Memory are properly captured using only 4 samples, which leads to an 80% saving of processing and communication, compared to the original 20 samples. From this saving (running the temperature management algorithm only 4 times instead of 20 times) we have to subtract the processing overhead at the sensor node; that is, $2 \times 20$ comparisons (i.e., single subtractions).

Furthermore, the trend of reports in Fig. 4 should not be overlooked. We can see, for example, in the case of $\Delta = 5°C$ (green-dashed-circle), after the initial report, no value is reported in the first $15ms$, whereas 3 values are reported between 15 and $30ms$. The reason being that, no drastic temperature changes happen in the first $15ms$, in contrast to the next $15ms$. We observe a similar trend in the $\Delta = 1°C$ case, which shows the asymmetric behavior of attention-based data collection. In other words, as intended, the frequency of reporting values, depends on the actual events observed instead of being uniformly distributed over time. This effect is more pronounced in the case of other signals where only one value is reported instead of 20 values (required by Nyquist criteria) over the $32ms$, and yet no event of importance is overlooked.

## V. Conclusion

In this paper, we propose a new architecture for data collection in MPSoC which is inspired by the concepts of self-awareness, more specifically, attention-based sensing. Sensors report a value to the management and processing unit only when an 'event of importance' has occurred. Interpretation of 'event of importance' is application dependent, in this study, that was a change of temperature bigger that a user-defined $\Delta$. To this end, an architecture, which uses a minimal hardware and processing overhead at the sensor node, can considerably reduce the communication and processing load at higher levels of the system. We evaluated the proposed method in a test scenario, where 10 sensors monitoring different units of a Nehalem architecture based processor were equipped with the attention-based data collection units. For the majority of the sensors, this led to more than 95% reduction in the number of reported values during a $32ms$ window, compared to the reported number of data, down-sampled using Nyquist criteria. Furthermore, we showed that by setting the attention span, $\Delta$, to $1°C$, a 35% saving on communication and temperature data processing can happen with virtually no loss of information. Whereas, by an attention span of $5°C$ an 80% improvement of efficiency can be achieved without missing any event of importance as far as temperature management is concerned.

## Acknowledgment

The authors would like to thank Professor Axel Jantsch for his extremely valuable input and constructive feedback.

## References

[1] S. Kosonocky, "ISSCC 2016 trends - digital architectures & systems," in *ISSCC, the International Solid-State Circuits Conference*, 2016.

[2] H. Esmaeilzadeh *et al.*, "Dark silicon and the end of multicore scaling," in *Proc. of the 38th ISCA*, pp. 365–376, 2011.

[3] A. M. Rahmani *et al.*, eds., *The Dark Side of Silicon*. Springer, 2016.

[4] K. C. Smith *et al.*, "Through the looking glass - trend tracking for isscc 2012," *IEEE Solid State Circuits Mag.*, pp. 4–20, 2012.

[5] S. Borkar, "Thousand core chips: a technology perspective," in *Proc. of the 44th DAC*, pp. 746–749, ACM, 2007.

[6] S. Furber, "Living with failure: Lessons from nature?," 2006.

[7] J. Keane and C. Kim, "An odometer for CPUs," *IEEE Spectrum*, vol. 48, pp. 26–31, May 2011. Online version appeared as "Transistor Aging".

[8] N. Dutt *et al.*, "Towards smart embedded systems: A self-aware system-on-chip perspective," *ACM TECS, Special Issue on Innovative Design Methods for Smart Embedded Systems*, vol. 15, no. 2, pp. 22–27, 2016.

[9] A. Jantsch and K. Tammemäe, "A framework of awareness for artificial subjects," in *Proc. of the CODES*, pp. 20:1–20:3, ACM, 2014.

[10] H. Hoffmann *et al.*, "A generalized software framework for accurate and efficient management of performance goals," in *Proceedings of the International Conference on Embedded Software*, pp. 1–10, Sept 2013.

[11] A. Chandra *et al.*, "Reference architecture for self-aware and self-wxpressive computing systems," in *Self-Aware Computing Systems: An Engineering Approach* (P. R. Lewis *et al.*, eds.), ch. 4, pp. 37–49, Springer, 2016.

[12] IBM Corporation, "An architectural blueprint for autonomic computing," 2006. IBM White Paper.

[13] N. TaheriNejad *et al.*, "Comprehensive observation and its role in self-awareness - an emotion recognition system example," in *Proc.of the FedCSIS*, 2016.

[14] J.-S. Preden *et al.*, "The benefits of self-awareness and attention in fog and mist computing," *IEEE Computer, Special Issue on Self-Aware/Expressive Computing Systems*, pp. 37–45, July 2015.

[15] J. Zeppenfeld *et al.*, "Applying ASoC to multi-core applications for workload management," in *Organic Computing - A Paradigm Shift for Complex Systems* (C. Müller-Schloer, H. Schmeck, and T. Ungerer, eds.), Autonomic Systems, ch. 5.3, pp. 461–472, Birkhäuser, 2011.

[16] E. Joint Conference on Knowledge-Based Software Engineering (7th : 2006 : Tallinn, E. Tyugu, and T. Yamaguchi, "Knowledge-based software engineering : proceedings of the seventh joint conference of knowledge-based software engineering," 2006. Conference held August 28-31, 2006 in Tallinn, Estonia.

[17] J. W. Valvano, *Introduction to Embedded Systems 1e*. Cengage Learning, 2010.

[18] P. A. Laplante, *Comprehensive Dictionary of Electrical Engineering*. Springer-Verlag Berlin Heidelberg, 1999.

[19] T. E. Carlson, W. Heirmant, and L. Eeckhout, "Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulation," in *2011 International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pp. 1–12, Nov 2011.

[20] W. Huang *et al.*, "Hotspot: a compact thermal modeling methodology for early-stage VLSI design," *IEEE Trans. on VLSI Sys.*, vol. 14, no. 5, pp. 501–513, 2006.

[21] H. Khdr *et al.*, "Power density-aware resource management for heterogeneous tiled multicores," *IEEE Trans. on Computers*, vol. 66, no. 3, pp. 488–501, 2017.