The transprecision computing paradigm: Concept, design, and applications

This is the post peer-review accepted manuscript of:

A. C. I. Malossi, M. Schaffner, A. Molnos, L. Gammaitoni, G. Tagliavini, A. Emerson, A. Tomás, D. S. Nikolopoulos, E. Flamand and N. Wehn, "The transprecision computing paradigm: Concept, design, and applications," 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, 2018, pp. 1105-1110. doi: 10.23919/DATE.2018.8342176

The published version is available online at: https://doi.org/10.23919/DATE.2018.8342176

The Transprecision Computing Paradigm: Concept, Design, and Applications

A. Cristiano I. Malossi
IBM Research – Zurich, Switzerland
acm@zurich.ibm.com

Michael Schaffner
ETHZ, Switzerland
schaffner@iis.ee.ethz.ch

Anca Molnos
CEA, France
anca.molnos@cea.fr

Luca Gammaitoni
Università di Perugia, Italy
luca.gammaitoni@nipslab.org

Giuseppe Tagliavini
Università di Bologna, Italy
giuseppe.tagliavini@unibo.it

Andrew Emerson
CINECA, Italy
a.emerson@cineca.it

Andrés Tomás
Universitat Jaume I, Spain
tomasan@uji.es

Dimitrios S. Nikolopoulos
Queen's University of Belfast, UK
d.nikolopoulos@qub.ac.uk

Eric Flamand
GreenWaves Technologies, France
eric.flamand@greenwaves-technologies.com

Norbert Wehn
University of Kaiserslautern, Germany
wehn@eit.uni-kl.de

***Abstract*—Guaranteed numerical precision of each elementary step in a complex computation has been the mainstay of traditional computing systems for many years. This era, fueled by Moore's law and the constant exponential improvement in computing efficiency, is at its twilight: from tiny nodes of the Internet-of-Things, to large HPC computing centers, sub-picoJoule/operation energy efficiency is essential for practical realizations. To overcome the power wall, a shift from traditional computing paradigms is now mandatory.**

In this paper we present the driving motivations, roadmap, and expected impact of the European project OPRECOMP. OPRECOMP aims to (i) develop the first complete *transprecision computing framework*, (ii) apply it to a wide range of hardware platforms, from the sub-milliWatt up to the MegaWatt range, and (iii) demonstrate impact in a wide range of computational domains, spanning IoT, Big Data Analytics, Deep Learning, and HPC simulations.

By combining together into a seamless design transprecision advances in devices, circuits, software tools, and algorithms, we expect to achieve major energy efficiency improvements, even when there is no freedom to relax end-to-end application quality of results. Indeed, OPRECOMP aims at demolishing the ultra-conservative "precise" computing abstraction, replacing it with a more flexible and efficient one, namely transprecision computing.

***Index Terms*—Approximate Computing; Inexact Computing; Energy-Efficiency; Low Power Computing; Architecture Design**

I. Introduction

In the last 10-years, the demand for new computing strategies driven by energy-efficiency has grown exponentially [1]. Flop-per-watt (thus, per-euro) has become de-facto a driving model in hardware design. Results in this direction have been significant [2], leveraging first multi-core parallelism and then recently moving toward heterogeneous architectures (e.g., multicore CPU coupled with GP-GPUs). However, these evolutions will not be sufficient in the long term. To maintain an exponential increase in computational efficiency, we rely either on an unlikely breakthrough discovery in hardware technology, or on a fundamental change in computing.

The project OPRECOMP (website: oprecomp.eu) acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the European Union's Horizon 2020 research and innovation programme, under grant agreement No 732631. IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. Other product and service names might be trademarks of IBM or other companies.

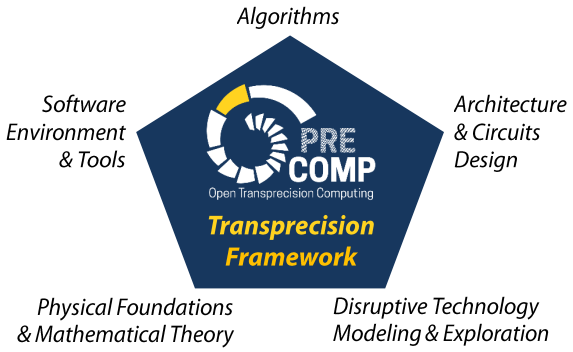

Fig. 1: OPRECOMP aims at providing a holistic *transprecision framework* spanning all layers of computing systems.

The H2020 European project OPRECOMP explores the latter opportunity, and puts a strong bet on *transprecision computing* rooted into the key intuition of exploiting approximation in hardware and software, from both a statistical and a deterministic viewpoint. The mission of OPRECOMP is to build demonstrators that prove that this idea holds (i) in a huge range of application scenarios in the domains of IoT, Big Data Analytics, Deep Learning, and HPC simulations, and (ii) from the sub-milliWatt to the MegaWatt range, spanning nine orders of magnitude.

To achieve this goal a truly holistic approach covering all the constitutive layers of complete computing systems must be followed and integrated in a vertically-orchestrated solution, as shown in Figure 1. Usable transprecision computing requires bidirectional cross-layer interaction. Real-life applications are strongly data-dependent, hence precision cannot be other than

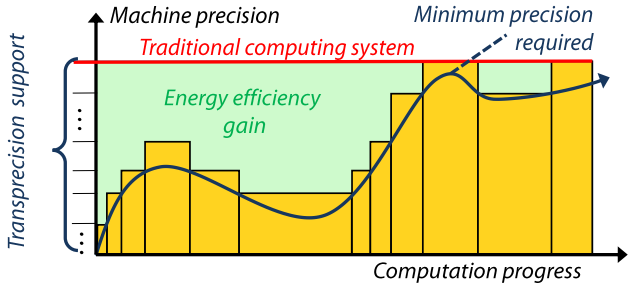

Fig. 2: *Transprecision computing* enables fine control over precision in space and time, thereby leading to significant energy savings without sacrificing overall quality of results.

a controlled open loop. A top-down flow of information is needed to modulate precision requirements in space and time and depending on instance-specific data. A bottom-up flow is also critical to ensure closed-loop adaptation of requirement, self-tuning and on-line learning.

This paper is organized as follows: in Section II we introduce and define the transprecision computing paradigm. Then in Sections IV, III, and V we show how such a concept is going to be realized through the OPRECOMP roadmap, putting together re-thinking of algorithms, new hardware design, and special software packages and environments. Then in Section VI we unveil first results and forecast impact on real applications. Concluding remarks are summarized in Section VII.

II. Transprecision Computing

Transprecision computing is rooted into the key intuition of exploiting approximation in both hardware and software (the former due to the presence of variability or unavoidable physical fluctuations, the latter due to computation with limited precision) to boost energy efficiency. While this is clearly tied to the rapidly developing research area known as approximate computing [3], [4], a transprecision computing framework goes beyond the state-of-the-art along several axes, and more precisely:

1) It controls approximation in space and time (when and where) at a fine grain though multiple hardware and software feedback control loops, see Figure 2.
2) It does not imply reduced precision at the application level, even though it is also possible to exploit application-level softening of precision requirements for extra benefits.
3) It takes inspiration from nature by defining computing architectures that operate with a smooth and wide range of precision vs. cost trade-off curve.

The key barrier to a widespread adoption of classic approximate computing is the lack of an application-to-hardware framework for managing precision without compromising application quality. More precisely, the lack of guarantees and of tight error control is the main showstopper. In a *transprecision computing framework* this limit is overcomed by a fine-grained and distributed control of hardware operation coupled with scalable, feedback based runtimes and a programming model enabling on-line tracking of error metrics and modulation of operating parameters to meet quality-of-results at the application level.

III. Architecture Design

In the OPRECOMP project, we aim to demonstrate advantages of a transprecision framework from the sub-mW to the MW range, spanning nine orders of magnitude. To achieve this goal, we investigate the fundamental physical aspects of computation, and develop two radically different hardware platforms that act as proxies to server-grade compute platforms and low-power IoT nodes, spanning the range from mW to kW. These demonstrators will be used to showcase our work, and the obtained measurements will be used to project the results to a hypothetical MW system.

A. Limits Imposed by the Laws of Physics

In any computational platform, each calculation is performed by a physical system. As such, the changes in the physical system determine the energetic cost associated with the act of computing. It is important to note that while some energetic costs are due to technological constraints and thus can be arbitrarily lowered, some others have fundamental bounds set by the laws of physics. As an example of fundamental cost is the bound set by any irreversible change in the system entropy. The most popular connection between computation and physical system was originally proposed by Landauer in 1961 [5]. Landauer's principle states that the operation of resetting a bit in any physical device generates a minimum amount of dissipated heat equal to $Q = -k_B T \ln 2$, where k_B is the Boltzmann constant and T the temperature of the device. While in a recent paper [6] it has been shown that computation with traditional logic gates, per se, can be performed in principle with an arbitrarily small amount of energy, the reset operation, typical of computing initialisation or memory writing, requires that a finite minimum amount of energy is consumed. Such an energy toll, however can be attenuated by trading precision in computation with potential energy saving [7]. In this project we will identify which costs are fundamental and which are technological, determining trade-offs between accuracy and cost of computation. The results will set the lower bound of energetic performances achievable in the project.

B. Targeted Platforms and Hardware Architecture

In order to leverage the full potential of the transprecision concept, fine grained control over arithmetic precision in compute units as well as bit error rates in the memory hierarchy is required. Hence, the level of required hardware support goes beyond what traditional hardware platforms provide today. To this end, OPRECOMP leverages the *Parallel Ultra Low Power* (PULP) platform [8], which is an open-source many core platform based on energy-efficient, 4-stage, in-order RISC-V processor cores [9]. The PULP platform will be extended with transprecision-capable memory hierarchy, transprecision accelerators and accelerated processing units (APUs) that are tightly integrated into the RISC-V cores. As illustrated in Figure 3, the transprecision-capable PULP platform will be used as a basis for two demonstrator systems that are optimized for two different power envelopes and which rely on the same transprecision capabilities:

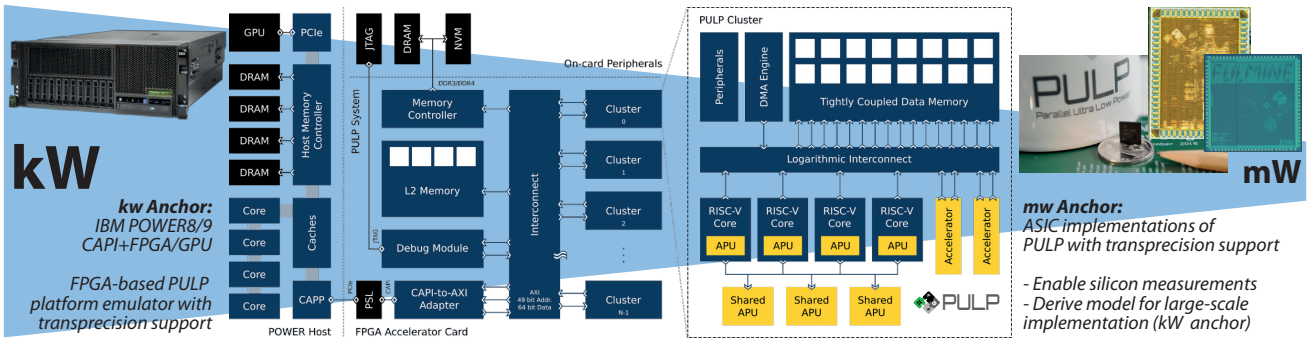

Fig. 3: The *Parallel Ultra Low Power* (PULP) platform will be extended with transprecision-capable computing units, accelerators and memory infrastructure, and serve as a basis for two demonstrator systems (kW and mW anchors).

1) mW Demonstrator – technology demonstrator for the transprecision units: The mW range demonstrator will be based on an eight-core PULP system geared towards achieving the highest possible energy efficiency, by making use of aggressive power saving features and improving the computing efficiency of individual cores. This base system will be extended to have transprecision capabilities in its interconnection, data computation and storage units. The complete platform will include the transprecision enhanced PULP system and include various sensors to demonstrate the validity of our approach in real applications developed using the software environment developed in this project.

2) kW Demonstrator – scalability and functional demonstrator for the software stack: Our second platform will be one node of an HPC system with transprecision capabilities. In a first phase, this system will be based on an existing POWER8 - CAPI (Coherent Accelerator Processor Interface) attached FPGA prototyping system developed by IBM. The POWER8 processor will be used for precise calculations, and an interconnected array of PULP based processing units with controllable precision will be implemented on the FPGA. In a second phase, we will upgrade the node to a POWER9 with CAPI 2.0 connection to an advanced version of PULP on a large-scale FPGA. This system will be used to validate the approach. The actual power and performance numbers will be extracted through measurements on a dedicated ASIC that will be manufactured containing an array of the PULP based transprecision computation cores. Finally, the results on the kW system will be projected up to a potential multi-node MW architecture.

C. Optimal Use of the Memory Subsystem is Crucial

It is well known that many applications are not compute but memory bandwidth limited. For instance, Google has recently shown that the Tensor Processing Unit (TPU) [10] is limited by memory bandwidth in four out of six neural network inference workloads. Thus, memories play a crucial role in advanced computer architectures. Today's most prominent external memories are Dynamic Random Access Memories (DRAMs). DRAMs largely contribute to the overall power consumption and compute performance. For instance, it has been shown that in natural language processing more than 80% of the energy consumption comes from the DRAMs and only 18% from the computing itself. A second challenge is DRAM performance – since DRAMs are optimized for density, their performance lags behind the performance of the compute units. Although the peak performance of DRAMs has increased strongly over the years due to improved internal data pre-fetching and interface performance, the periodically required refresh and the timing dependencies can result in up to 10× access time variations.

In transprecision computing, we address the memory challenge from several directions for both platform anchors (kW and mW). First, the concept of approximate computing is extended to DRAMs [11]. In approximate DRAMs the refresh is lowered or even completely switched-off and this can result in possible data errors [12]. Thus, the use of approximate DRAM depends largely on the DRAM retention error behavior, data lifetimes and application robustness as described in Figure 4.

Second, we integrate and exploit application knowledge in the memory subsystem composed of the memory controller and the DRAMs. Memory controllers are orchestrating the DRAM accesses from the compute fabric to the DRAMs itself. The controller is a complex device that is, in contrast to compute architectures, still general purpose with limited context information. The context information is restricted to the buffer size. Here, we follow an application specific memory controller

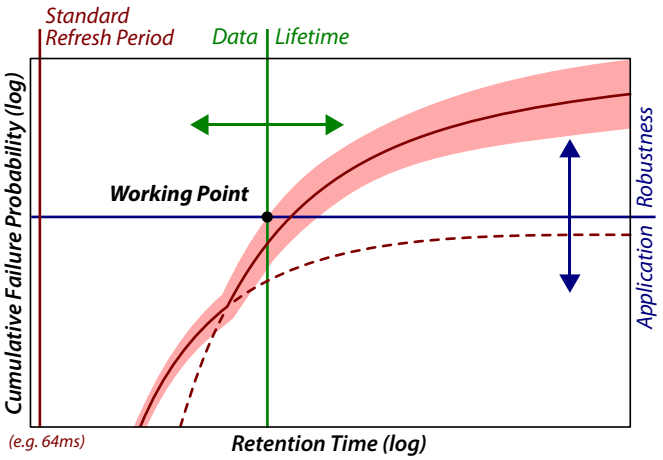

Fig. 4: Qualitative Retention Error Behavior

approach: for a given application specific access pattern, the memory controller can be configured in a way such that the data are optimally mapped into the DRAM banks/ranks/channels. An optimal mapping minimizes the row misses and, hence, maximizes the bandwidth and energy efficiency. Third, we use special power down and refresh strategies that can be tuned to the application characteristic. We limit our optimizations not only to DRAMs but also to emerging memories, such as Resistive RAMs.

IV. Algorithm Design and Error Propagation

Transprecision can be applied to the algorithms both at the low level, introducing approximation in a subset of operations, as well as at the high level, changing the algorithm workflow and reducing the complexity of the operations in selected parts of the computation. The first requires deep understanding of number format behaviour, the latter is more linked with algorithm-rethinking. In both cases, new metrics have to be introduced to assess quality, performance, and energy-savings. Moreover, error propagation must be carefully controlled.

A. Quality Metrics

Quality metrics (QMs) are selected to (i) test the correctness of the system and measure the balance between performance gains versus quality loss, (ii) assess the results over the full life span of the project and range of architectures, and (iii) enable closed-loop monitoring with low-weight check overhead. Most often, the choice of QMs will depend on the target algorithm, micro-benchmark and application. QMs include performance metrics (PMs) and quality-of-the-result metrics (QRMs).

Basic PMs are direct performance indicators such as (execution)-time, energy consumption and, for power-constrained systems, power dissipation. From these direct PMs, it is possible to compose several derived metrics such as the energy consumption-per-work unit. A popular example for a derived metric is the FLOPS/Watt (floating-point arithmetic operations per second and Watt, that is, operations per Joule), promoted by the Green500 ranking [2]. For parallel systems, two additional relevant PMs are strong and weak scalability.

QR metrics (QRMs) assess the quality of the response in terms of accuracy. QRMs compare, whenever possible, the approximate solution (ApS) and the exact solution (ExS) of the process, to quantify the accuracy (i.e., QR) of the former. In the case when the ExS is not available, QRMs should help reach a conformable decision based on the ApS. At this point, it is worth mentioning that the threshold to qualify an ApS as acceptable is highly dependent on the application.

B. Computer Arithmetic

Floating point arithmetic is the standard for scientific applications. Its main advantage over integer arithmetic is that with a relatively small number of bits, it is able to represent a large range of numbers with a small relative error. Floating point numbers $y = \pm m \times \beta^{e-t}$ have three integer components: sign $\pm$, mantissa (significand) m and exponent e, while the base β is implicit and fixed for all numbers (usually $\beta = 2$). The sign is stored in one bit and the mantissa is maintained as an integer with t bits in the range $0 \leq m \leq \beta^{t-1}$.

The accuracy of any floating point format can be specified by the smallest number $\epsilon = 2^t$ that, when added to one, yields a result different from one. The maximum relative error when representing a number using round to nearest is the unit round-off $u = \epsilon/2$. Following this model, the result of any basic arithmetic operation with neither overflow nor underflow is

$$fl(x \text{ op } y) = (x \text{ op } y)(1 + \delta_{x \text{ op } y}) \qquad |\delta_{x \text{ op } y}| \leq u,$$

with op $\in \{+, -, \times, \div\}$ and $\delta_{x \text{ op } y}$ is the relative error.

Fixed point exploits an integer representation to store fractional numbers, where the number of bits dedicated to the fractional and non-fractional depend on the scaling factor s. A number y is stored as an integer i following the relation $y = \pm i/s$. The main advantage of fixed point over floating point is that it does not need specific hardware to run efficiently, and the representation can be easily adapted in software by simply changing the scaling factor. Usually this factor is a power of two allowing to use binary shifts for intermediate operations. The representation precision when using round to nearest is $u = 1/2s$.

The main drawback of fixed precision is its very limited range compared to a floating point representation that uses a similar number of bits. The result of an operation in fixed point without overflow can be modelled as

$$fx(x \text{ op } y) = x \text{ op } y + \Delta_{x \text{ op } y} \qquad |\Delta_{x \text{ op } y}| \leq u\,,$$

with op $\in \{+, -, \times, \div\}$ and $\Delta_{x \text{ op } y}$ is the absolute error. The relative error can be as large as 1 for the smallest representable numbers – this is different from floating point where the relative error is constant for all numbers except denormals.

C. Error Propagation

Multiplication, division and addition are benign operations in floating point representation; that is, the error in the operation result is of the same order of magnitude as that present in the operands. However, error in subtraction can be large due to cancellation, as a result of the loss of significant digits when subtracting two similar numbers. For fixed point the case is just the opposite: addition and subtraction are benign operations while multiplication and division can incur large errors. Algorithms should be designed to avoid cancellation in floating point and overflow/underflow in fixed point arithmetic. In general, when using different formats for each operand, the error in the operation result will be as large as the largest representation error from any of the operands.

V. Software Environment for Transprecision

The OPRECOMP application and system software stack have three key objectives. The first is to offer programmers the abstractions for expressing structure and disciplined approximation of data during computation or communication. The second is to hide the complexity of a full transprecision computing hardware and system software stack from users, while addressing both architectural and intrinsic hardware heterogeneity. The

third is to leverage the full range of approximate computing technologies available in the OPRECOMP hardware stack.

To achieve these objectives OPRECOMP follows four methodological approaches:

- The use of dynamic code generators to produce algorithmic kernels with variable vector and scalable type depth, that use variable precision arithmetic or replace floating point arithmetic with integer approximation on platforms that lack floating point support. We use runtime code generation methods that dynamically adapt the precision of floating point computation to either improve program performance or reduce energy consumption.
- The introduction of new number formats requiring a reduced number of bits (i.e., reducing precision and dynamic range) to increase the energy efficiency of hardware platforms. Floating-point operations are a major contributor to the energy consumption of platforms, due to circuit complexity and data movements (e.g., memory-memory, memory-register). In particular, in the OPRECOMP project we analyse the benefits provided by extended floating-point type systems, including:
  - a technique to emulate non-IEEE floating-point formats on development environments;
  - a methodology to collect statistics (time and power) related to floating-point operations;
  - machine learning techniques for understanding the relation between variable precision and output error;
  - an abstraction layer to integrate different tools for fine-grain tuning of variable precision.
- The extension of standard programming languages and standards (i.e., C, OpenMP) with abstractions to express transprecision arithmetic in a manner that is composable, dynamic and context-sensitive. The extensions include abstractions to map computation to specific arithmetic units, map data to heterogeneous memory subsystems with capabilities for data compression and control memory system reliability to enable performance and power optimisation.
- A Hardware Abstraction Layer (HAL) and associated operating system to leverage approximation in communication, data transfers, and error control and recovery. We provide scheduler extensions to expose affinity for hardware units with variable precision and perform accuracy-aware workload allocation. We explore an exact serving scheduler as an evaluation baseline and a distributed swarm-based scheduler, where small-scale allocation and accuracy tuning decisions are taken based on simple probabilistic rules. We further provide region-based memory allocation strategies for data that can be approximated in memory and automatic page allocation and migration policies between approximate and non-approximate regions, considering data access patterns and memory access timing. Finally, we introduce a storage and IPC stack with efficient compression and decompression paths used on demand.

The system software stack of OPRECOMP is complemented with new hardware monitoring capabilities. The software stack interfaces to novel memory controllers that gather information about data access patterns, lifetime, access latency, accuracy, precision, criticality and reliability. Additionally, the OPRECOMP system software stack leverages low cost monitoring and corrective functions implemented in hardware. These include low cost error confinement schemes implemented for specific instructions.

TABLE I: OPRECOMP micro-benchmark overview.

	Micro-Benchmarks	Platform mW	Platform kW	Power8 Baselines Serial	Power8 Baselines OpenMP	Power8 Baselines CUDA
DL	CNN	○	○	✓	✓	✓
	GD	–	○	✓	✓	✓
	PCA	○	○	✓	✓	○
	BLSTM	○	○	✓	✓	○
Big Data	PageRank	○	○	✓	✓	✓
	K-Means	○	○	✓	✓	○
	SVM	○	○	✓	○	–
	KNN	○	○	✓	✓	○
HPC	GLQ	○	○	✓	✓	–
	FFT	○	○	✓	○	–
	Stencil	○	○	✓	✓	✓
	SparseSolve	○	○	✓	○	○

✓ Implemented ○ Planned – Not envisioned

VI. Applications

OPRECOMP aims at demonstrating transprecision benefit on a wide range of applications. To do that we follow a micro-benchmark strategy, inspired by the Dwarves [13], [14]: the first list of micro-benchmarks includes 12 kernels that are key for a wide variety of real-world applications in the domains of Deep Learning, Big-Data and Data Analytic, High Performance Computing (HPC) and Scientific Computing. The set of micro-benchmarks covers, regular (sequential, single-threaded CPU) code, OpenMP parallelized code and NVIDIA CUDA implementations of GPU kernels, as shown in Table I. We identify at least three different applications from each of the three domains and provide a stand-alone baseline implementations of those. We believe that almost all of them are suitable for execution of the targeted mW- and kW-platform, when the input dataset for the benchmarks is scaled accordingly. In what follows, we provide a short overview of the current micro-benchmarks, as well as of the transprecision computing techniques that we will exploit during the project.

A. Deep Learning

Over the last few years, the utilisation of deep learning methods to real problems has grown significantly in many commercial and industrial domains. To reflect this trend, we include four common algorithmic patterns from this field in our benchmark list: Convolutional Neural Network (CNN), Stochastic Gradient Descent (SGD), Bidirectional Long-Short-Term Memory (BLSTM) and Principal Component Analysis (PCA). The BLSTM and CNN micro-benchmarks are two commonly used neural network configurations, for analyzing one-dimensional time sequences, e.g., audio, and two-dimensional data, e.g., pictures or individual video frames. Furthermore, neural networks have to be trained on a large data set prior

to deployment, and such training is typically achieved using SGD algorithms. Finally, PCA is a common step encountered in many learning and signal processing algorithms, e.g., for example to reduce the dimensionality.

B. Big-Data and Data Analytics

In this domain, the goal of many applications is to analyze data to identify trends or extract specific features. In some cases the data analysis requires a significant amount of computation, in which case it is performed in the cloud (the kW platform). In other cases, the goal is to take immediate action based on the observed data (e.g. in robotics/drones, self-driving cars). In this case, local embedded processing near the sensor (the mW platform) is preferred due to privacy or security concerns, or limitations in the communication bandwidth. For OPRECOMP we chose four well-known and largely used algorithms in this field covering different tasks: unsupervised classification (k-means), supervised classification (KNN and SVM) and web search (Page Rank).

C. HPC and Scientific Computing

Numerical simulation is nowadays an essential tool for scientific and technological research, widely utilized in industry to model all sorts of chemical and physical processes. To cover this domain, we have selected four computational patterns present in many HPC/scientific computing applications: Gauss-Legendre Quadrature (GLQ), Fast Fourier Transform (FFT), Stencil and Sparse Solve. The GLQ solves the 1D numerical integration problem by approximation, and it is typically employed in solutions based on Finite-element methods (FEMs). The FFT is one of the most important algorithms in computational science and is widely employed in many applications. Stencil algorithms update array elements according to a fixed pattern and are common in scientific and engineering applications, e.g., computational fluid dynamics. Finally, we selected the Conjugate Gradient (CG) method as a representative Sparse Solve case study, as CG exhibits a data access pattern that is common to all other solvers of this family, and, furthermore, it has been recently proposed to complement the popular LINPACK benchmark to rank the TOP500 computing systems.

D. Transprecision Computing Optimisations

The benchmark suite contains micro-benchmark kernels that are used in applications where the output quality of the final results might be still adequate or good-enough even if approximate concepts (e.g. less precision arithmetic) are used for computing them. Those kernels allow us to explore aggressive approximation techniques which can be translated into power and/or performance gains.

The exploration space of the transprecision concept addresses the question on how algorithms can be built to run and take advantage of four main directions:

1) *Data types*, especially low precision versions such as half float (16 bit) or non-standard 8bit floats.
2) *Interaction with Transprecision Memory*. Memory systems consume a large part of the system power, and refresh cycles are major contributor to this power. Lowering refresh rates allows naturally to save power, however the read-out error statistic of such a system would change.
3) *Iterative Transprecision Concepts*. Iterative algorithms (e.g., the conjugate gradient, page rank) allow transprecision concepts to be explored in the case where no additional error is permitted at the end of the computation. The self-correcting property of iterative algorithms with transprecision concepts might introduce temporary errors during the early iterations that can be corrected towards the end of the computation where precision requirements are gradually increased to meet the final requirements.
4) *General Algorithmic Changes*. Any general high-level change of the algorithm can be explored, however, we note that such optimizations are application-specific.

VII. Conclusions

The long term scientific goal of OPRECOMP is to develop concepts, methods, hardware and software building blocks for practical transprecision computing systems. The foundational character of the project comes from the key idea that approximation needs to become a degree of freedom and a controllable design parameter as opposed to a "necessary evil" as it is seen in traditional computing systems. OPRECOMP takes a multi-scale system design approach based on open architectures. We aim to achieve at least a one order of magnitude improvement in energy efficiency demonstrating that the transprecision idea holds in a wide range of application scenarios in the domains of IoT, Big Data Analytics, Deep Learning, and HPC simulations.

References

[1] "ICT-Energy Strategic Research Agenda," Report, 2016.

[2] "The Green500 list," June 2017, http://www.green500.org.

[3] Q. Xu, N. Kim, and T. Mytkowicz, "Approximate Computing: A survey," *IEEE Design and Test*, 2016.

[4] S. Mittal, "A survey of techniques for approximate computing," *ACM Computing Survey CSUR*, 2016.

[5] R. Landauer, "Irreversibility and heat generation in the computing process," *IBM journal of research and development*, vol. 5, no. 3, pp. 183–191, 1961.

[6] M. Lopez-Suarez, I. Neri, and L. Gammaitoni, "Sub-kbt micro-electromechanical irreversible logic gate," *Nature communications*, vol. 7, 2016.

[7] L. Gammaitoni, "Computing with uncertainty can help lowering the minimum energy required?" *NANOENERGY Letters*, 02/2013 2013. [Online]. Available: www.nanoenergyletters.eu

[8] D. Rossi *et al.*, "A 60 GOPS/W, -1.8 V to 0.9 V Body Bias ULP Cluster in 28nm UTBB FD-SOI Technology ," *Solid-State Electronics*, 2016.

[9] M. Gautschi *et al.*, "Near-Threshold RISC-V Core With DSP Extensions for Scalable IoT Endpoint Devices," *TVSLI*, 2017.

[10] N. P. Jouppi *et al.*, "In-Datacenter Performance Analysis of a Tensor Processing Unit," in *Proceedings of ISCA '17*. New York, NY, USA: ACM, 2017, pp. 1–12.

[11] M. Jung *et al.*, "Invited: Approximate computing with partially unreliable dynamic random access memory: Approximate DRAM," in *ACM/EDAC/IEEE DAC*, June 2016, pp. 1–4.

[12] ——, "A Platform to Analyze DDR3 DRAM's Power and Retention Time," *IEEE Design Test*, vol. 34, no. 4, pp. 52–59, Aug 2017.

[13] K. Asanovic *et al.*, "The landscape of parallel computing research: A view from berkeley," EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2006-183, Dec. 2006.

[14] E. L. Kaltofen, "The "Seven Dwarfs" of Symbolic Computation," in *Numerical and Symbolic Scientific Computing*, ser. Texts & Monographs in Symbolic Computation, U. Langer and P. Paule, Eds. Springer Vienna, 2012, pp. 95–104.