Opportunities for Energy Efficient Computing: A Study of Inexact General Purpose Processors for High-Performance and Big-data Applications

Peter Düben
AOPP, University of Oxford
Oxford, United Kingdom
dueben@atm.ox.ac.uk

Parishkrati, Sreelatha Yenugula, John Augustine
Indian Institute of Technology Madras
India
parishkratihhh@gmail.com,
sreelathayenugula@gmail.com, augustine@cse.iitm.ac.in

K. Palem
Electrical Engineering and Computer Science
Rice University, Houston, USA
kvp1@rice.edu

Jeremy Schlachter
EPFL
Lausanne, Switzerland
jeremy.schlachter@epfl.ch

Christian Enz
EPFL
Lausanne, Switzerland
christian.enz@epfl.ch

T. N. Palmer
AOPP, University of Oxford
Oxford, United Kingdom
t.n.palmer@atm.ox.ac.uk

Abstract—*In this paper, we demonstrate that disproportionate gains are possible through a simple devise for injecting inexactness or approximation into the hardware architecture of a computing system with a general purpose template including a complete memory hierarchy. The focus of the study is on energy savings possible through this approach in the context of large and challenging applications. We choose two such from different ends of the computing spectrum—the IGCM model for weather and climate modeling which embodies significant features of a high-performance computing workload, and the ubiquitous PageRank algorithm used in Internet search. In both cases, we are able to show in the affirmative that an inexact system outperforms its exact counterpart in terms of its efficiency quantified through the relative metric of operations per virtual Joule (OPVJ)—a relative metric that is not tied to particular hardware technology. As one example, the IGCM application can be used to achieve savings through inexactness of (almost) a factor of 3 in energy without compromising the quality of the forecast, quantified through the forecast error metric, in a noticeable manner. As another example finding, we show that in the case of PageRank, an inexact system is able to outperform its exact counterpart by close to a factor of 1.5 using the OPVJ metric.*

I. Introduction and Overview of Findings

Inexact or (approximate) computing has been presented as a significant and potentially unorthodox approach to realizing low energy computing systems. In terms of hardware manifestations and models, following the early foundational work that identified this opportunity (please see [8], [28], [29]; and [30], [31] for a more comprehensive overview), the most robust validations have been through construction of specific hardware artifacts such as adders, multiplier or FFT engines [22], [23], [20], [21]. For additional references on inexact hardware, architectures and related topics, please also see [1], [32], [12], [39], [35], [2]. More recently, this work has also been extended to evaluations of large-scale applications—in the context of our own group, climate modeling and weather prediction served as the driving example in which the floating point operations were rendered inexact [9].

It is however widely recognized that if we were to consider general purpose computing systems—as opposed to architectures with accelerators with a high degree of customization [5]—energy costs associated with the memory portion of the computation dominates to a great extent. As a result, work that isolates and identifies opportunities in the computational part of the application by largely overlooking the memory component, as opposed to treating it as a significantly dominant component, is justifiably criticized whenever *a traditional architectural template involving a data path and a standard memory hierarchy is the platform of choice.*

In this paper, we remedy this situation by extending the reach of our studies to understand the limits to energy benefits that can be gleaned at the application level by including inexactness *both in the computation along the data path and also in the memory system.* We do this by considering two applications, the Intermediate General Circulation Model (IGCM) for weather and climate modeling [4] and the widely used PageRank algorithm [2],[26] at the heart of internet search. In terms of understanding the limits to opportunities, the IGCM model involves significant amounts of floating

978-3-9815370-4-8/DATE15/©2015 EDAA

point computation whereas, in contrast, PageRank is significantly memory bound; thus, they lean towards two ends of the opportunity spectrum.

In order to evaluate the opportunity due to inexactness from the perspective of energy savings, we introduce an *energy model* (cf. Section II) which is based on two input elements. The first element involves an application *activity profile* involving activity as the program executes in different parts of the architecture. Our activity profiles are relative with the activity in the entire application adding up to 100%. These activity profiles were gathered using Valgrind [36], which is a publicly available tool. Next, we define the *normalized energy cost* model where the simplest integer operation represents the cheapest operation quantified as 1 unit of energy consumption—all of the other activities are then represented as multiples of this value. The overall *energy cost* is then estimated by combining the activity profile with the normalized energy cost in an obvious way once the total operation count is known.

Continuing, we consider two scenarios for executing our two applications. The first scenario involves architectural elements—adders, multipliers, DRAM main memory, SRAM caches—to be exact namely embodying full precision. In the second scenario, both the computation elements on the data path as well as the memory structures are inexact—we vary and increase degrees of inexactness. Of the mechanisms possible, in this paper, we induce inexactness by reducing the number of bits in the mantissa of the double precision floating point number or the width of integer operations. For example, we use between 8 and 12 bits for the significand or mantissa when considering the atmosphere application and consider 14 and 16 bits when evaluating the PageRank application. In both cases, the fully precise or exact case will have 52 bit mantissas. The *quality* of the solution as the inexactness is increased is modeled as follows: in the IGCM model, as the level of inexactness is increased, the forecast error of a meteorological quantity is used to quantify the "goodness" of the solution. In PageRank, we use three different distance measures—*variation distanc*e or VD [24], the ℓ_∞ distance, and Kendall's τ distance [18]—from a baseline exact execution.

In all cases, we estimate the energy savings as the application is executed using an exact model, compared to increasingly inexact versions. As a summary, when considering the IGCM model (Section IV), with no detectable change in the quality of the prediction quantified through the *global mean error metric,* we could achieve over a factor of three in energy savings. Similarly, for PageRank, the baseline is an execution using a parameter called tolerance of 0.01. With this value, the page ranking computed on an exact system is deemed to be the "golden value" computed using full precision floating point values with a 64 bit representation has the ideal average VD value of 0. Let us now consider an inexact version with 26 bits. The computation can be realized with a fraction of 0.68 of the virtual energy compared to the exact case and the associated VD value goes up to 1.005. If we tried to achieve the same level of energy gains using an exact computing system, our choice was to increase the tolerance parameter, which lowers the number of operations performed. Surprisingly, with an energy of 0.89 of the exact case, the average VD value jumped up to a whopping 100.3—almost two orders of magnitude larger than the comparable inexact configuration!

In achieving energy savings through inexactness, it is reasonable to consider simplifying the application by altering its program implementation either statically, or by lowering some execution parameter such as the number of its iterations for example. Thus, the gains need not have been derived from inherent system technology improvements but rather purely through algorithm or program changes only. To sharply isolate the efficiencies that our approach gives us which can be attributed to inexactness from the architecture elements, we introduce a novel metric which we will refer to as *operations per virtual Joules (OPVJ).*

We show that a reduction of precision can increase OPVJ in the atmosphere model by more than a factor of three at approximately the same "goodness" of the solution—to reiterate, inexactness is induced by lowering the number of bits in the Mantissa. We see a very similar retention of goodness in the solution in PageRank along with an increase in OPVJ value by a factor of 1.5 for the exact and inexact cases compared in the previous paragraph.

II. Modelling Processor Energy Consumption

With the intent of gaining insights that are oblivious to the exact technology in use—be it current or future—our goal in this work is to show the relative gains offered by inexact computing. Nevertheless, we wish to rigorously model energy across all major activities in both computation and memory systems. Towards this goal, against each type of activity that we address, we assign an energy cost relative to the least expensive activity, which, in our case is integer operations pegged at 1 virtual Joule (vJ). Every other activity, therefore, is assigned a relative cost in virtual Joules; see Table 1. These relative costs have been derived as best estimates obtained by insights from source [19].

We classify computational operations broadly into floating point operations, integer operations, and others (that would include branches/control flow, etc.)[1].

Category	Relative Energy (vJ)
Cache access	15
Main memory	30
Floating point	2
Integer	1
Branches (others)	1

Table 1. List of operations along with associated relative energy costs in virtual Joules.

[1] Caches were designed to significantly reduce access times (sometimes up to a factor of 100), but their energy costs are closer to that of main memory – a factor of two in our accounting. The implication, of course, is that any effort to exploit inexactness techniques that can be applied to both types of memory accesses will pay rich dividends.

The energy consumed by an execution in vJ is determined by combining the activity profile-number of operations in each category—with the relative energy as a weighted sum.

III. Methodology for Deducing Energy Gains

For each application that we consider, we measure the relative number of operations or memory accesses (denoted by c_i); from each category i listed in Table 1. Let e_i denote the relative energy consumed per operation/access of category i when executed in an exact manner; this is column 2 in Table 1. Then, our total relative energy in virtual Joules for exact computation is given by:

$$T_{exact} = \sum_i c_i \times e_i.$$

To measure the gains, we then measure the fraction (f_i) of operations/accesses in each category that are executed in an inexact manner. Typically, activities that involve data can be made inexact, but all control statements, branches, iterators, etc. must be retained exactly. As in [7], our architectural model supports some activities to be inexact—including inexact memory banks, FPUs and others, while the rest are exact. Let e_i^* (typically $\ll e_i$) denote the reduced energy (in virtual Joules) consumed by the inexact execution of an operation/access in category i; arriving at suitable e_i^* values will be discussed shortly. We thus get the total energy for inexact computation as a linear combination given by:

$$T_{inexact} = \sum_i c_i(f_i e_i^* + (1 - f_i)e_i).$$

This implies that our energy cost relative to the cost expended in exact computation is $T_{exact}/T_{inexact}$, which we simply call the *relative energy gains*.

While there are many tools and literature for measuring e_i values, we have to take special care in measuring e_i^* values. To reiterate, in this work, inexactness is introduced by truncating the least significant bits. Thus our energy savings are functions of the number of truncated bits. In the specific case of the floating point unit, any inconsistency in the exponent would result in a huge error distance, thus, truncation is only applied to elements computing the mantissa. Both, exact and truncated arithmetic units are synthesized in the UMC 65 nm process, following the standard digital design flow. Since the estimated power consumption is strongly dependent on the technology, the energy cost of each inexact operation is normalized using the standard 64-bit computational blocks as reference. To be sure that the relative gains are resulting from truncation, each inexact variant of a design is synthesized under the same delay and area constraints than the exact counterpart.

In the context of memory, we used a publicly available tool, DRAMsim [37], to determine energy estimates of the main memory. Again, these numbers are normalized to the cost of a standard 64-bit read or write operation. Energy savings are obtained by reducing the number of bits written or read, per operation. Since we allow the least significant bits to be present in memory, the energy for refreshing remains the same as for exact memory; this was modelled faithfully. We assume that the cache is built out of SRAM cells, and following [25], the cache, energy consumption scales linearly with the width of the word[2]. It has to be noted that the relative savings for memory may however be slightly optimistic since we do not account for the instruction codes memory access.

IV. Case Study Of IGCM for Climate and Weather Modeling

A. An atmosphere model

We studied the use of inexact hardware in the Intermediate General Circulation Model (IGCM) that represents atmospheric dynamics in global simulations [4], [16], [17], [34]. IGCM is used to solve the governing fluid dynamical equations for atmospheric motions in three dimensions on the sphere. In contrast to an operational weather or climate model, it does not include a representation of components such as physical tracers, biosphere, ocean, topography, water vapor or clouds. However, IGCM represents the basic building block of some of the most important weather and climate models (such as IFS and ECHAM) and provides a meaningful testbed for operational models. The model can run in climate configurations for long time intervals (from years to centuries). In this paper, we consider short term simulations equivalent to weather forecasts for several days.

Numerical simulations of the atmosphere are of crucial importance for reliable forecasts of weather and climate. The quality of these forecasts is dependent on the resolution used, and complexity of the numerical models, which is limited by the computational power of today's supercomputing facilities.

B. Analysis of the Results

Earlier studies [10], [11] have reported that a reduction in precision does not cause a catastrophic reduction of the quality of model simulations for many applications in atmospheric modelling. These studies focused on relating error to the number of bits in the mantissa but did not consider the relationship to energy, which we now pursue in this paper. As long as certain parts of the model are left with high precision (in specific the dynamics of large-scale pattern and the time-stepping scheme), numerical precision can be reduced heavily with no strong increase in model error. The robustness of atmosphere models in the presence of inexactness can be explained by the inherent uncertainty that is present in model simulations mainly due to physical processes that cannot be fully captured, and the high viscosity which needs to be used for turbulent closure.

Model runs are performed in a standard testbed for three dimensional simulations of the atmosphere, the so-called Held-Suarez configuration, with 20 vertical levels. If resolution is increased, grid spacing is decreased and quality of model simulation will be consequently improved. We calculate the model error by comparing runs at coarser

[2] We note that overheads involving possible dynamic alteration of width such as gating are not included in our estimates, and in this regard, our results should be viewed as optimistic estimates if a dynamically adjustable architecture is to be considered.

resolution to a simulation with much higher resolution with a grid spacing of 125 km. Typically, the different simulations will diverge from the high-resolution simulation with time, due to model errors. We perform a forecast with 235 km resolution in an exact setting using full double precision that serves as a reference or baseline for the model's quality. We will than reduce computational cost by using either lower resolution (260 km and 315 km) and double precision, or by using the same resolution as the reference run (235 km) but inexact hardware.

Model Run	Normalized Energy Demand	Forecast error day 2	Normalized operations per virtual Joule
235 km, 64 bits	1	2.3	1.00
260 km, 64 bits	0.82	2.8	1.01
315 km, 64 bits	0.47	4.5	1.02
235 km, 24 bits	0.35	2.3	2.83
235 km, 22 bits	0.32	2.3	3.09
235 km, 20 bits	0.29	2.5	3.41

Table 2. Normalized energy consumption, global mean error for geopotential height in [m] at day 2 (averaged over 5 forecasts) and normalized operations per virtual Joule (OPVJ) for the different simulations where the forecast error results were presented in [10] and [11].

Table 2 provides values for the normalized energy demand, the forecast error in *geopotential height* at day 2 of the forecast and the normalized number of operations per virtual Joule (OPVJ). Geopotential height is a standard meteorological diagnostic that relates pressure to height. The simulations with inexact hardware clearly show a reduced normalized energy demand and an increased number of operations per virtual joule. It is found that all simulations with 235 km resolution clearly show a smaller forecast error in comparison to simulations in double precision at lower resolution (260 km or 315 km), even if floating point precision is truncated at 20 bits in the floating point representation (8 bits in the significand). As reiterated in the table, the forecast error with inexact hardware is hardly affected in a significant way in comparison to the double precision simulation.

V. Case Study of the PageRank Algorithm: A Memory Intensive Big Data Application

A. The Pagerank Algorithm

The web has played a pivotal role in bringing out the need to build applications that scale well. Therefore, our choice of a memory intensive application (and the underlying algorithm) to showcase the power of inexactness is, quite naturally, one that seeks to answer an important question at the heart of web search: how to determine as to which pages are important? The PageRank algorithm [6], [24] developed in 90's was a revolutionary idea that redefined web search simply because it answered the question of ranking pages in a clever, elegant, and scalable manner. Going beyond the interesting history that surrounds its invention and early use, this algorithm has turned into a quintessential "big data" application [3], [14].

The main aim of the PageRank algorithm is to deduce a score (called PageRank value) for each web page using the underlying link structure. The Web is seen as a graph in which the vertices are web pages and a directed edge exists from one page p_1 to another p_2 if there is a hypertext link in p_1 pointing to p_2. In the normalized sense, the PageRank value of a page p is—without addressing all subtleties—the probability with which a random surfer (who goes from one page to a neighbor chosen uniformly at random) will be at p in the steady state. Although PageRank is defined in this Markovian manner, it is often implemented in a deterministic manner in which each node starts off with a weight of 1 and the weights are iteratively distributed to the neighbors. This iterative process will converge (when changes to the PageRank values is below a specified tolerance parameter) to values that are (close to being) in proportion to the stationary distribution of the random surfer described earlier.

Since the PageRank values are in proportion to probabilities in the stationary distributions, we use the following natural distances that are commonly used:

- The first metric we use is the variation distance [24], which is defined as $\frac{1}{2}\sum_{p\in\mathcal{P}}\left|\pi_p - \pi_p^*\right|$, where the summation is over the set $\mathcal{P}$ of all web pages, and π_p (resp., π_p^*) denote the steady state probability of the random surfer being at page p computed using an inexact implementation (resp., exact implementation). This is proportional to the ℓ_1 distance between the two distributions.
- The second metric we use is the ℓ_∞ norm (also called the max norm) and is defined as the $\max_p |\pi_p - \pi_p^*|$.
- Inspired by the purpose of ranking the importance of web pages for which PageRank algorithm was designed, we also measure the Kendall's τ distance, which is defined as the number of inversions between the two rank orders (i.e., exact and inexact) of the web pages induced by sorting them based on their PageRank values.

PageRank is primarily used to process very large web graphs to ensure that important web pages are placed ahead of unimportant ones in web search results. This naturally makes it robust against mildly inexact results. For example, if the exact and the inexact system differ only by a small value ϵ in the ℓ_∞ norm, then, no two pages that are more than 2ϵ apart in PageRank value will ever be wrongly ordered in the inexact system. While the ℓ_∞ norm captures this worst case behavior, the other two norms are a bit more forgiving in the sense that they are sums of differences and thereby may not reveal situations where a few web pages processed in the inexact system have very different PageRank values (or very different rank orderings) from when processed exactly.

B. Analysis of Results

In our first experiment, we have shown the amount of error we incur when we introduce inexactness via bitwidth truncation. Our error was computed with respect to a baseline implementation of PageRank in which the tolerance was set to

0.01 without any inexactness (i.e., no truncation). In order to compute the number of times different activities were performed, we had to write our own PageRank code in which the algorithm was self-contained without any reference to third party libraries; to ensure correctness of our implementation, we tested our code against the PageRank implementation in Graphlab [13]. We used Valgrind [36] to profile the execution to arrive at the number of executions of each type of activity.

As in the case of the IGCM study, we were able to retain an almost-error-free execution despite the significant level of inexactness that was injected with associated energy savings. In order to understand these savings better, we compared them against naïve approaches such as those that alter the tolerance in PageRank. Some of these findings are summarized in Figure 1 where we have mapped out the number of operations per virtual Joule as a function of the number of iterations of PageRank to reach a given tolerance threshold. We notice that the number of operations per virtual Joule increases by factor of 1.5 or so, as we introduce inexactness—thus implying that we get more computation done per unit of energy expended. Consequently, as we go from a tolerance of 0.04 (exact) to a value of 0.01 (inexact) and for a comparable energy budget, the PageRank algorithm is able to iterate more in the latter case and thus achieve lower error levels relative to the golden solution.

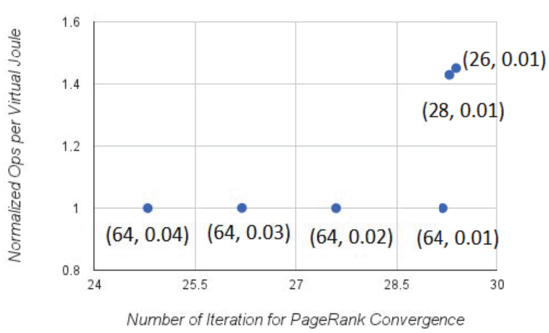

Figure 1. The relationship between exact and inexact realizations of PageRank.

VI. Concluding Remarks

The work described in this paper represents a preliminary study of the opportunities present in achieving energy savings through novel approaches such as inexactness. Much remains to be done, in particular, through energy and power models rooted in hardware measurements (such as those in [22]). One interesting general conclusion that we could draw from the results in this paper is that inexactness has an interesting disproportionate benefit as follows, true for both of our applications. Inexactness might lower the quality of an application by a certain amount but in return for energy gains. Now, if we reinvest these gains in a different part of the application—computing the IGCM at a finer scale or increasing the number of iterations of PageRank—the overall result could in fact be better than with the original exact solution that is typically viewed as a gold standard. We have data that justifies this in our own preliminary studies reported here and intend to expand along this path in a technical report and a future paper. However, this insight opens up a novel style of co-design and also room for automating this through efficient design space exploration and compiler support for achieving energy efficiencies through inexactness. It is also likely that machine learning and control-theory as well as traditional approaches could play a crucial role in approaching such an exploration [38], [33], [15]. Additionally, our approach to studying energy (and by extension power consumption) through a relative virtual metric might be of broader and more general interest. Finally, more ambitious forms of inexactness beyond precision variation have shown to yield better results in the context of other applications in earlier work, and exploring the value of such methods in the comprehensive models described here will also be of value. We are aware that we did not include considerations of computing speed or performance in that sense in the present paper since we wanted to entirely focus on the energy consumption aspect. However, we do have estimates for the relationship to simultaneous speed or running time improvements which will be a part of our follow-up report along with more robust modeling and validations.

Acknowledgment

Peter Düben and Tim Palmer evaluated variation of inexactness and performed model simulations. They also evaluated the power consumption for the atmospheric application. Jeremy Schlachter and Christian Enz evaluated the relative power savings of single operations, through the use of inexact hardware and in the case of the memory component, in collaboration with John Augustine, Parishkrati, Sreelatha Yenugula. The latter group also implemented the PageRank algorithm and instrumented the code and conducted the experiments that led to its analysis. They also defined suitable parameters to evaluate error in inexact PageRank. Krishna Palem developed the power model, served as the bridge between the error analysis and energy modeling, and coordinated the work of writing the paper. Finally, we are greatly indebted to Avinash Lingamneni for useful discussions and suggestions. We are also thankful to V. Nandivada Krishna for pointing us to Valgrind.

Peter Düben and Tim Palmer are funded by an ERC grant (Towards the Prototype Probabilistic Earth-System Model for Climate Prediction, ERC project number 291406).

References

[1] Amant, R. S., Yazdanbakhsh, A., Park, J., Thwaites, B., Esmaeilzadeh, H., Hassibi, A., ... & Burger, D. (2014). General-Purpose Code Acceleration with Limited-Precision Analog Computation. In Proceedings of the 41st Annual International Symposium on Computer Architecture (ISCA)

[2] G. P. Arumugam, P. Srikanthan, J. Augustine, K. Palem, E. Upfal, A. Bhargava, Parishkrati, S. Yenugula. Novel Inexactness Aware Algorithm Co-design for Energy Efficient Computation. To appear in DATE 2015.

[3] Bahman Bahmani, Kaushik Chakrabarti, and Dong Xin. 2011. Fast personalized PageRank on MapReduce. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data* (SIGMOD '11). ACM, New York, NY, USA, 973-984.

[4] Blackburn, M., 1985: Program description for the multi-level global spectral model.

[5] Shekhar Borkar and Andrew A. Chien. 2011. The future of microprocessors. Commun. ACM 54, 5 (May 2011), 67-77. DOI=10.1145/1941487.1941507

[6] Brin, S. and L. Page, "The anatomy of a large-scale hypertextual Web search engine," Computer Networks and ISDN Systems, Volume 30, Issues 1–7, April 1998, Pages 107-117, ISSN 0169-7552.

[7] Chakrapani, L. N., Korkmaz, P., Akgul, B. E., & Palem, K. V. (2007). Probabilistic system-on-a-chip architectures. ACM Transactions on Design Automation of Electronic Systems (TODAES), 12(3), 29.

[8] Cheemalavagu, S., Korkmaz, P., & Palem, K. V. (2004, September). Ultra low-energy computing via probabilistic algorithms and devices: CMOS device primitives and the energy-probability relationship. In Proc. of The 2004 International Conference on Solid State Devices and Materials (pp. 402-403).

[9] Düben, P. D., Joven, J., Lingamneni, A., McNamara, H., De Micheli, G., Palem, K. V., & Palmer, T. N. (2014). On the use of inexact, pruned hardware in atmospheric modelling. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 372(2018), 20130276.

[10] Düben, P.D., McNamara, H., Palmer, T.N. (2014). The use of imprecise processing to improve accuracy in weather & climate prediction, Journal of Computational Physics, Volume 271, Pages 2-18, ISSN 0021-9991.

[11] Düben, P.D. and T. N. Palmer (2014). Benchmark Tests for Numerical Weather Forecasts on Inexact Hardware. Mon. Wea. Rev., 142, 3809-3829.

[12] Esmaeilzadeh, H., Sampson, A., Ceze, L., & Burger, D. (2012, December). Neural acceleration for general-purpose approximate programs. In Proceedings of the 2012 45th Annual IEEE/ACM International Symposium on Microarchitecture (pp. 449-460). IEEE Computer Society.

[13] http://graphlab.org/ Accessed on Dec 7, 2014.

[14] Hartmut Klauck, Danupon Nanongkai, Gopal Pandurangan, Peter Robinson, "Distributed Computation of Large-scale Graph Problems," to appear in SODA 2015.

[15] Hoffmann, H. (2014, July). CoAdapt: Predictable Behavior for Accuracy-Aware Applications Running on Power-Aware Systems. In Real-Time Systems (ECRTS), 2014 26th Euromicro Conference on (pp. 223-232). IEEE.

[16] Hoskins, B. J., and A. J. Simmons, 1975: A multi-layer spectral model and the semi-implicit method. Quart. J. Roy. Meteor. Soc., 101, 637–655, doi:10.1002/qj.49710142918.

[17] James, I. N. and J. P. Dodd, 1993: A simplified global circulation model.

[18] M. G. Kendall, A New Measure of Rank Correlation, *Biometrika*, Vol. 30, No. 1/2 (Jun., 1938), pp. 81-93.

[19] Li, S., Ahn, J. H., Strong, R. D., Brockman, J. B., Tullsen, D. M., & Jouppi, N. P. (2009, December). McPAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures. In Microarchitecture, 2009. MICRO-42. 42nd Annual IEEE/ACM International Symposium on (pp. 469-480). IEEE.

[20] Lingamneni, A., Enz, C., Palem, K., & Piguet, C. (2013). Synthesizing parsimonious inexact circuits through probabilistic design techniques. ACM Transactions on Embedded Computing Systems (TECS), 12(2s), 93.

[21] Lingamneni, A., Enz, C., Palem, K., & Piguet, C. (2013). Designing energy-efficient arithmetic operators using inexact computing. Journal of Low Power Electronics, 9(1), 141-153.

[22] Lingamneni, A., Enz, C., Palem, K., & Piguet, C. (2014, September). Highly energy-efficient and quality-tunable inexact FFT accelerators. In Custom Integrated Circuits Conference (CICC), 2014 IEEE Proceedings of the (pp. 1-4). IEEE.

[23] Lingamneni, A., Basu, A., Enz, C., Palem, K. V., & Piguet, C. (2013, May). Improving energy gains of inexact DSP hardware through reciprocative error compensation. In Design Automation Conference (DAC), 2013 50th ACM/EDAC/IEEE (pp. 1-8). IEEE.

[24] Mitzenmacher, M. and E. Upfal, Probability and Computing: Randomized Algorithms and Probabilistic Analysis. Cambridge University Press, 2005.

[25] Okuma, T., Cao, Y., Muroyama, M., & Yasuura, H. (2002). Reducing access energy of on-chip data memory considering active data bitwidth. In Low Power Electronics and Design, 2002. ISLPED'02. Proceedings of the 2002 International Symposium on (pp. 88-91). IEEE.

[26] L. Page, S. Brin, R. Motwani, T. Winograd, The PageRank Citation Ranking: Bringing Order to the Web, unpublished manuscript, 1998.

[27] Palem, K., & Lingamneni, A. (2013). Ten years of building broken chips: the physics and engineering of inexact computing. ACM Transactions on Embedded Computing Systems (TECS), 12(2s), 87.

[28] Palem, K. V. (2003). Computational proof as experiment: Probabilistic algorithms from a thermodynamic perspective. In Verification: Theory and Practice (pp. 524-547). Springer Berlin Heidelberg.

[29] Palem, K. V. (2003, October). Energy aware algorithm design via probabilistic computing: from algorithms and models to Moore's law and novel (semiconductor) devices. In Proceedings of the 2003 international conference on Compilers, architecture and synthesis for embedded systems (pp. 113-116). ACM.

[30] Palem, K., & Lingamneni, A. (2013). Ten years of building broken chips: the physics and engineering of inexact computing. ACM Transactions on Embedded Computing Systems (TECS), 12(2s), 87.

[31] Palem, K., Lingamneni, A., Enz, C., & Piguet, C. (2013, September). Why design reliable chips when faulty ones are even better. In ESSCIRC (ESSCIRC), 2013 Proceedings of the (pp. 255-258). IEEE.

[32] Ranjan, A., Raha, A., Venkataramani, S., Roy, K., & Raghunathan, A. (2014, March). ASLAN: synthesis of approximate sequential circuits. In Proceedings of the conference on Design, Automation & Test in Europe (p. 364). European Design and Automation Association.

[33] Rubio-González, C., Nguyen, C., Nguyen, H. D., Demmel, J., Kahan, W., Sen, K., ... & Hough, D. (2013, November). Precimonious: Tuning assistant for floating-point precision. In Proceedings of SC13: International Conference for High Performance Computing, Networking, Storage and Analysis (p. 27). ACM.

[34] Simmons, A. J., and B. J. Hoskins, 1975: A comparison of spectral and finite-difference simulations of a growing baroclinic wave. Quart. J. Roy. Meteor. Soc., 101, 551–565, doi:10.1002/qj.4971014291.

[35] Shanbhag, N. R., Abdallah, R. A., Kumar, R., & Jones, D. L. (2010, June). Stochastic computation. In Proceedings of the 47th Design Automation Conference (pp. 859-864). ACM.

[36] http://valgrind.org/. Accessed on Dec 7, 2014.

[37] David Wang, Brinda Ganesh, Nuengwong Tuaycharoen, Kathleen Baynes, Aamer Jaleel, and Bruce Jacob. 2005. DRAMsim: a memory system simulator. *SIGARCH Comput. Archit. News* 33, 4 (November 2005), 100-107.

[38] Xin Sui, Andrew Lenharth, Don Fussell, Keshav Pingali. Tuning variable-fidelity programs. Submitted for publication.

[39] Ye, R., Wang, T., Yuan, F., Kumar, R., & Xu, Q. (2013, November). On reconfiguration-oriented approximate adder design and its application. In Proceedings of the International Conference on Computer-Aided Design (pp. 48-54). IEEE Press.