Software Optimizations for Reduced Energy Consumption: A Feasibility Study

Stephen I. Roberts
and Stephen Jarvis Department of Computer Science
The University of Warwick Chris January
and Jonathan Byrd Allinea Software
TODO

***Abstract*—Reducing energy consumption is a prerequisite for future advances in computing at scale. This fact is driving performance engineers to investigate software level power optimization as a means to reduce the energy consumed by scientific codes. This paper presents an investigation into the feasibility and potential benefits of this approach. We assess various power measurement and modelling techniques for their ability to help developers identify power optimizations. We then attempt to provide a realistic idea of how much benefit can be expected as a result of applying these optimizations. We show that for current hardware there is limited scope for improving energy efficiency through software optimisation alone.**

INTRODUCTION

Driven by Moore's Law, advances in processor design have delivered improvements in CPU performance for decades. As physical limits are reached, however, refinements to the same basic technologies are beginning to show diminishing returns. One side-effect of this is an unsustainable rise in system power usage, which the US Department of Energy has identified as a primary constraint for exascale systems [2].

Hardware manufacturers are already prioritising energy efficiency in their processor designs [1]. In turn, some groups have suggested that software modifications will be required to fully exploit the energy efficiency improvements of modern processors [3]. This is analogous to the current practice of tuning code for reduced runtime by exploiting specific processor features like vectorisation or cache hierarchy. These groups expect targeted optimisation to be applied to reducing power consumption in the future.

Code optimisation is a complex task during which developers typically rely on the support of a range tools and techniques including profilers and performance models. Up until now the overarching goal of performance engineers has been to minimize run time. The tools which have emerged over the years have therefore focussed on a specific class of optimization - namely code transformations which speed up execution. An expanded tool set will be required if the kind of multi-objective optimization necessary to encompass both power and runtime is to become commonplace.

A body of research is accumulating as the search for techniques to identify and reason about software power optimizations continues. TODO: Paragraph stating that work has commenced building up techniques to do this. Note a lot will be covered in prior art.

- Measurement vs modelling
- Power is the integral and hard to measure

The remainder of this paper is organized as follows. Section I gives an overview of prior research carried out in this area. Section **??** then provides a commentary about how the techniques described in Section I may be applied practically.

Sec **??**

I. REVIEW OF PRIOR ART

Prior art has by and large focussed on power modelling. We present a taxonomy of prior art in this field

TODO:

- Measurement vs modelling
- Power is the integral and hard to measure
- Approaches to measurement
- Modelling taxonomy

II. POWER PROFILING

$$P_{dyn} \propto CV^2Af \tag{1}$$

$$P_{leak} \propto V\left(ke^{\frac{-qVth}{ak_aT}}\right) \tag{2}$$

Baseline against which to compare the work of others.

<fragment>Accuracy figures without context are notoriously unreliable. To compensate for this we compare the outputs of various models against a baseline we have devised. This baseline consists of what we regard as the simplest non-trivial power model conceivable. This model stands in as a sort of null hypothesis test, our justification being that a complex model only adds value to the extent with which it outperforms this toy model.</fragment>

<fragment>Our toy model is not the simplest model possible - It is well established and readily apparent that runtime is the largest contributory factor to power consumption. One could therefore imagine a simple power model</fragment>

<fragment>We consider this to be the absolute minimum power consumption possible.</fragment>

<fragment>Two components to our investigation. Firstly, the upper bound imposed by the baseline power consumption. Secondly, as we can only view power figures approximately, the error introduced into these models necessarily

limits their usefulness as optimization tools beyond a certain point.</fragment>

III. Power Optimization

Having shown that

TODO: Decompose the model - find baseline vs non-baseline components TODO: This kind of shows us that the baseline dominates

<fragment>Put another way, any optimization which trades runtime for power has a limited window of</fragment>

TODO: equationify fact that energy is power * time, and assume we have power decreases, time static or increases TODO: equationify baseline lt optimized lt unoptimized lt roofline TODO: note - roofline is tdp max

TODO: Do maths think - by what margin would power have to go down to justify longer runtime? ratio of cost per watt, amortized cost per second

Nothing discussed so far precludes power optimization in practice. TODO: imply limits thus far are theoretical Even these tight limits may still admit some benefits at extreme scale. Our final argument however is strictly economic. Re-word: A great deal of attention is paid to the fact that power costs are approaching parity with machine construction costs. The {implicit, unspoken} {consequence, corollary, implication} being that this has not yet happened. TODO: Ultimate point being here the price difference, machine vs power cost places a further limit on optimization utility. Even if we manage to find a slower, more power efficient method of computing a given result, the cost of energy saved has to be less than the added amortized runtime cost.

IV. Results

V. Conclusions

TODO: IN an intuitive sense the covariance between these two signals, namely power and time consumption, is prohibitively high. The window of optimization exists in the ...word meaning freedom/decoupled component/... window

References

[1] Nasser Kurd, Muntaquim Chowdhury, Edward Burton, Thomas P Thomas, Christopher Mozak, Brent Boswell, Manoj Lal, Anant Deval, Jonathan Douglas, Mahmoud Elassal, et al. 5.9 haswell: A family of ia 22nm processors. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International*, pages 112–113. IEEE, 2014.

[2] John Shalf, Sudip Dosanjh, and John Morrison. Exascale computing technology challenges. In JoséM.LaginhaM. Palma, Michel Daydé, Osni Marques, and JoãoCorreia Lopes, editors, *High Performance Computing for Computational Science – VECPAR 2010*, volume 6449 of *Lecture Notes in Computer Science*, pages 1–25. Springer Berlin Heidelberg, 2011.

[3] Y.S. Shao and D. Brooks. Energy characterization and instruction-level energy model of intel's xeon phi processor. In *Low Power Electronics and Design (ISLPED), 2013 IEEE International Symposium on*, pages 389–394, Sept 2013.