Energy Reduction of Loops using DVFS and Duty Cycle Modulation

***Abstract*—Dynamic Voltage and Frequncy Scaling (DVFS) and Duty Cycle Modulation (DCM)are two {complimentary?} techniques for putting processors into low power mode. In this report, we propose to apply different DVFS and DCM settings to loops within an application according to their characterization. Different loops of the same application exhibit different charateristics and may prefer different settings. By applying the "best" power level setting to each loop, energy savings could be achieved. Our experiments show that we can reduce energy consumption of applications by applying "right" DCM setting for the "right" loops.**

I. Introduction

Dynamic Voltage and Frequency Scaling (DVFS) and Duty Cycle Modulation (DCM) are two power saving techinques on Intel Sandy Bridge processors. DVFS saves power by reducing the frequency and voltage supply of the chip. Duty Cycle Modulation reduces power consumption by freezing the cores for a specified time interval – the effect is that the machine frequency is reduced (without changing the voltage). DVFS benefits (in terms of energy) coarse-grained loops more than fine-grain loops, because it has larger transition overhead than DCM. Duty Cycle Modulation has fast transition but it only affects frequency – the power savings might not be as significant as DVFS reduced to the same frequency. Because of the faster transition, DCM has more potential in controlling fine-grained code regions than DVFS.

In order to optimize applications for energy, we experimented with the community detection application in presence of frequency scaling and duty cycle modulation. We found that running the program at a lower frequency yielded the minimum EDP (energy-delay product) without being the minimum energy, as shown in Figure 1. We thought it might be due to memory behaviour of the application.

Then we looked at memory counters and found that this counter can in part determine a frequency (or duty cycle modulation level) that will lead to the best energy. We also tried to characterize loops and correlate the characteristics to the Frequency/ClockModulation setting. We hope that by running different loops at different low power mode we can achieve a minimum Energy (E) or Energy Delay Product (EDP).

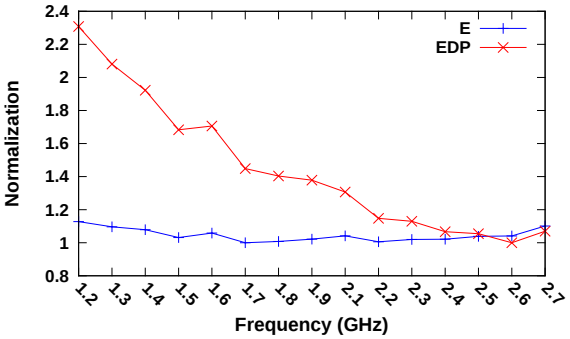


Fig. 1. Graph showing the normalized Energy(E)/EDP by varying the frequency. The minimum E/EDP is chosen as the baseline.

II. Benchmarks and Experimental Setup

In this work, we evaluated community detection code (graph clustering) for understanding of the best energy gear setting relating to different loop characters. The tests ran on a machine with dual socket 8-core Intel Xeon E5-2680 processor with 20MB (40MB total) L3 cache. The input graph is pnnl.bin, with which the whole application finishes in about 22 seconds. It contains three (among many) most significant loops. Each of the loops takes about 3 seconds.

III. Experimental Results

Figure 2-4 shows that three different loops inside the same application achieve minimum energy at different frequency scaling setting and duty cycle modulation setting. For example, in Figure 2(a), the minimum energy is achieved by operating the processsor at a frequency 1.6GHz. In Figure 2(b), the minimum energy is achieved by setting duty cycle modulation level to be 9, i.e. let the cores be active for 56.25% (=9/16) of the time.

We then let the three loops run with the frequency setting of 1.6GHz, 2.4GHz, 2.7GHz respectively and recorded the time and energy of the hybrid version. We also ran the three loops with the duty cycle modulation level of 9, 15, 16 respectively.

Table 1 compares the two runs with the minimum energy runs achieved by coarse-grained DVFS and DCM (i.e. presetting a fixed DVFS/DCM level and running the whole program with the preset configuration). We can see that coarse-grained DVFS achieved the minimum

energy, the fine-grained DVFS consumed the maximum energy. Fine-grained DCM reduced energy consumption compared with coarse-grained DCM. This table tell us that DCM can be used in fine-grained loops and DVFS can be more effective than DCM for coarse-grained (large) loops.

TABLE I
TIME AND ENERGY COMPARISON OF OPTIMAL CONFIGURATION TO BASELINE OF POLYBENCH KERNELS ON SANDYBRIDGE

Metrics	*Coarse-Grain*		*Fine-Grain*	
	DVFS	DCM	DVFS	DCM
Time	26.47	24.54	26.86	26.74
Energy	1647	1701	1728.23	1682
Power	62.23	69.33	64.34	62.80

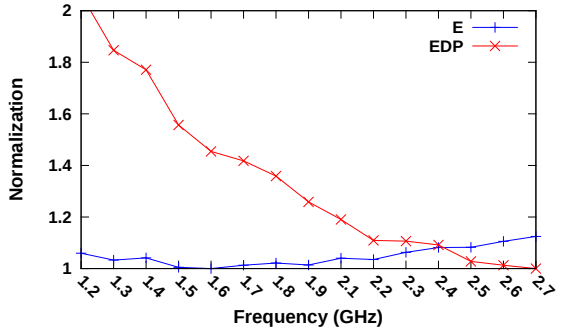
IV. RELATED WORK

Reducing Energy Usage with Memory and Computation-Aware Dynamic Frequency Scaling characterizes applications using static analysis and runtime tracing that automatically acquires application signatures - characterizations of the patterns of execution of each loop in an application. The characterization is matched with a set of benchmark loops, which have been fully explored for optimal frequency setting.

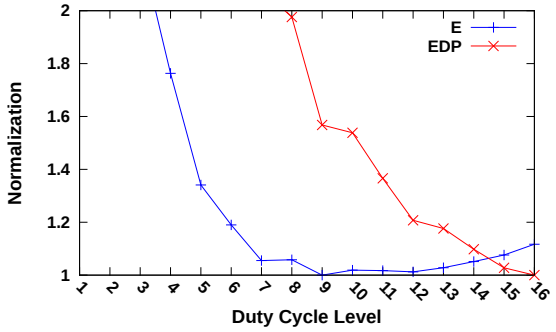
It would be good if we can test Alexandra's Decoupled Access-Execute model, where they called for faster DVFS transition and *modelled* faster DVFS for fine-grained access phase. We should test whether Duty Cycle Modulation is a good alternative to waiting for DVFS transition to be close to instantaneous.

V. TODO

TODO: Issues: 1) PNNL.bin tested is a small input. Even this small input brought significant run-to-run variations. 2) We did not achieve much energy savings in Table 1. 3) Initial test of some larger input makes one loop of the three loops dominant. Our approach is not applicable to applications with a dominant loop. 4) Larger input has even more significant run-to-run variance. We cannot tell whether the increase of execution time is due to energy setting or randomness brought by concurrency.

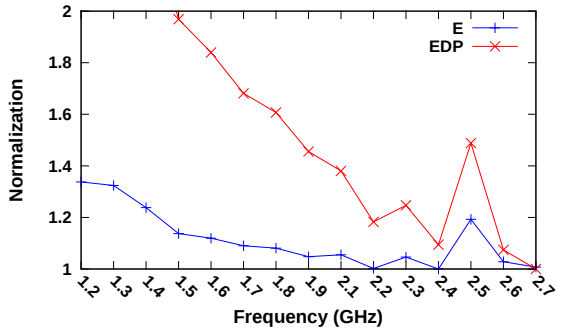


(a) Frequency Scaling Energy and Energy-Delay Product (Normalized) of the Loop

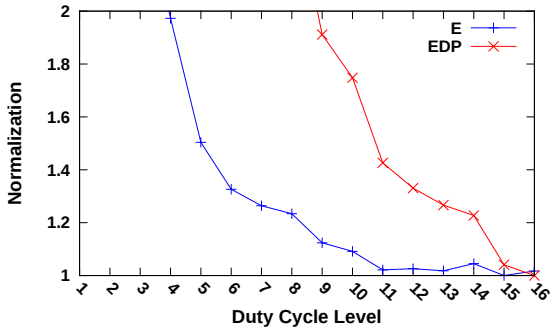


(b) Duty Cycle Modulation Energy and Energy-Delay Product (Normalized Values over 2 are Omitted) of the Loop

Fig. 2. Graph showing the energy (E) and energy-delay product (EDP) of the 1st loop by varing frequency and changing duty cycle modulation. Minimal E are achieved at 1.6GHz for DVFS and level 9 (56.25% active) for DCM. Minimal EDP are achieved at maximum settings for both DVFS and Duty Cycle Modulation. These minimal E and EDP are the baselines for normalization.

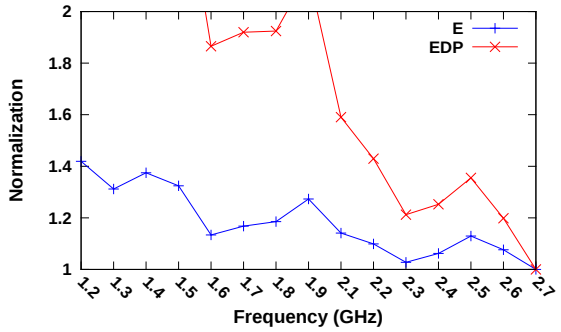


(a) Frequency Scaling Energy and Energy-Delay Product (Normalized Values over 2 are Omitted) of the Loop

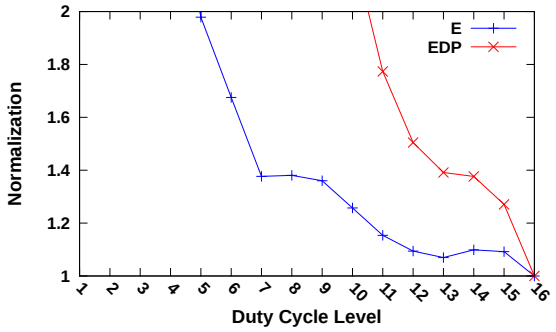


(b) Duty Cycle Modulation Energy and Energy-Delay Product (Normalized Values over 2 are Omitted) of the Loop

Fig. 3. Graph showing the energy (E) and energy-delay product (EDP) of the 2nd loop by varing frequency and changing duty cycle modulation. Minimal E are achieved at 2.4GHz for DVFS and level 15 (93.75% active) for DCM. Minimal EDP are achieved at maximum settings for both DVFS and Duty Cycle Modulation. These minimal E and EDP are the baselines for normalization.



(a) Frequency Scaling Energy and Energy-Delay Product (Normalized Values over 2 are Omitted) of the Loop



(b) Duty Cycle Modulation Energy and Energy-Delay Product (Normalized Values over 2 are Omitted) of the Loop

Fig. 4. Graph showing the energy (E) and energy-delay product (EDP) of the 3rd loop by varing frequency and changing duty cycle modulation. Minimal E and minimal EDP are at maximum settings for both DVFS and Duty Cycle Modulation. These minimal E and EDP are the baselines for normalization.