DVFS with Parallel Execution

Taejoon Song
ts693@cornell.edu

Goal

Our goal is to save energy by setting the DVFS with parallel execution using predicted execution time [1] for interactive applications (there is a constraint for the applications, see the section below).

Our contribution can be as follows:

1. Save more energy than previous DVFS work [1] which has the fixed deadline
2. Can predict execution time efficiently and accurately (not depend on specific algorithm or specific platform)
3. Or if we can somehow know the input for future jobs, this can be applied for other applications as well. (ex : repetitive input patterns, streaming applications)

Motivational Examples

$E = Pt$

$P \propto V^2 f,\ V \propto f^2,\ P \propto f^3$

$E \propto f^3$

No DVFS					
time	d/4	-	d/2	-	d
frame	A	slack	B	slack	C
freq	f	-	f	-	f
energy	(1/4)E	-	(1/2)E	-	E
total energy	(7/4)E				
noramlized energy	100				

Fixed deadline with DVFS [1]			
time	d	d	d
frame	A	B	C
freq	f/4	f/2	f
energy	(1/64)E	(1/8)E	E
total energy	(73/64)E		
noramlized energy	65		

DVFS with parallel exeuction			
time	d	d	d
frame (Proc 0)	A	C	
frame (Proc 1)	B		-
freq	f/4	f/4	(3/4)f
energy (Proc 0)	(1/64)E	(1/64)E	(27/64)E
energy (Proc 1)	(1/64)E	(1/64)E	idle E (assume it's 0)
total energy	(31/64)E		
noramlized energy	35		

(where `f` is maximum frequency, `d` is deadline, and `E` is energy when frequency is `f` and execution time is `d`)

Say we have two processors (Proc 0 and Proc 1) and frequency is shared between two processors.

Given the assumption that we can predict execution time perfectly for three frames in advance and that there is no overhead (migration/slice/dvfs) at all, total normalized energy for no DVFS, fixed deadline with DVFS, and DVFS in parallel execution are 100, 65, and 35, respectively.

Assumption (and Constraints)

1. We can predict execution time for the job using control flow and inputs.
2. We have the input stream not only for the current frame, but also for a number of future frames, so that we can predict the execution time for multiple frames beforehand.
3. We have multiple frames which have no dependency from each other.
4. We have enough memory spaces for a buffer to save pre-output.
5. We have enough slack time for predicting multiple frames, saving pre-output to the buffer, and setting the DVFS.

References

[1] Daniel Lo, Taejoon Song, and G. Edward Suh. “Prediction-Guided Performance-Energy Trade-off for Interactive Applications”, To appear in Proceedings of the 48th Annual International Symposium on Microarchitecture (MICRO), December 2015.