Fine-Grained Dynamic Voltage and Frequency Scaling (DVFS) for Precise Energy and Performance Tradeoff

Based on the Ratio of Off-Chip Access to On-Chip Computation Times

Kihwan Choi, Student Member, IEEE, Ramakrishna Soma, Student Member, IEEE, and Massoud Pedram, Fellow, IEEE

I.INTRODUCTION

Presented by Gamlath G.R.U.Y.- 180191H

❑ **What is DVFS Technique?**

❑ **Why DVFS is important?**

❑ **DVFS with General Purpose OSs**

❑ **DVFS technique in this paper**

❑ **Main Contributions**

- ➢ First actual implementations of an intraprocess DVFS policy.
- ➢ Regression model is proposed to approximately determine the CPU idle time.
- ➢ Evaluation of the proposed method is performed through actual hardware measurements for a number of different applications.

II. PRIOR WORK

❑ Previous DVFS-related works;

1. Coarse-grained
2. Fine-grained

❑ Techniques used in previous works;

- ➢ Multitask scheduling in the Operating Systems
- ➢ Intra task voltage-scheduling technique
- ➢ Method based on a software feedback loop
- ➢ Checkpoint-based algorithm
- ➢ Compiler-assisted DVFS techniques
- ➢ Intra task DVFS for multimedia application
- ➢ Intra task scheduling method
- ➢ Microarchitecture-driven DVFS technique

III. PERFORMANCE-ENERGY TRADEOFFS

Presented by Jayalath M.W.K.S - 180260U

A. Workload Partitioning

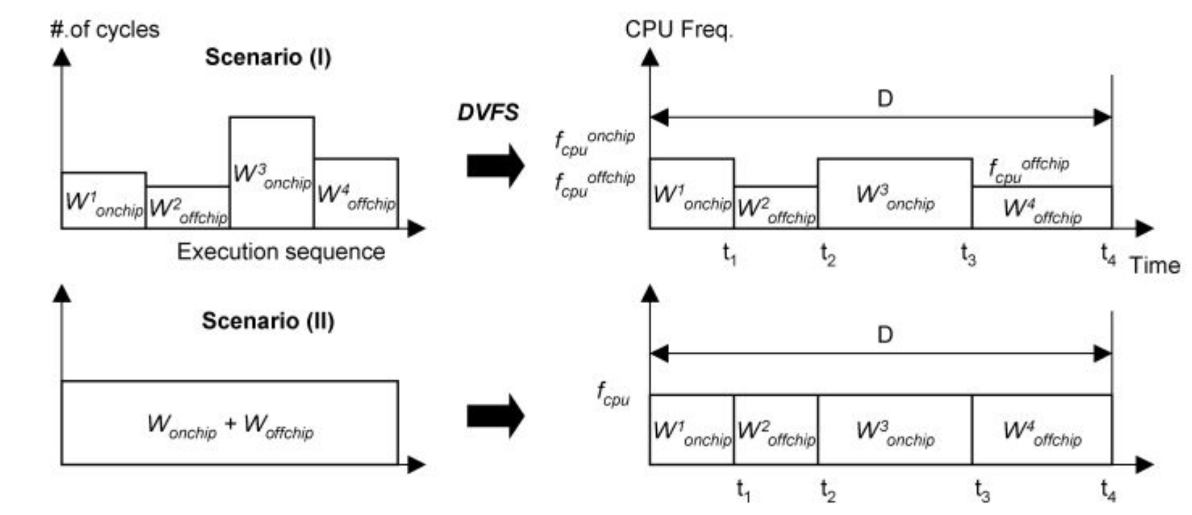
- ➢ What does workload partitioning mean?
- ➢ What is the end goal?

W_{onchip} - **On chip workload c. cycles**

$W_{offchip}$ - **Off Chip workload c.cycles**

Case1: sequence known

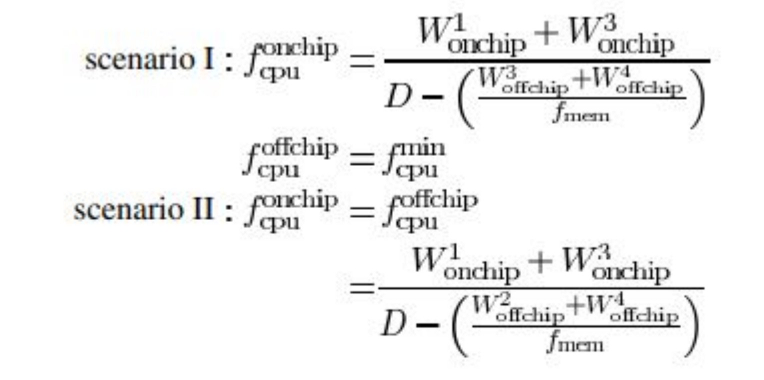
Case2: sequence unknown



Setting the CPU frequency:

Restrictions and conditions:

1. Off chip workload be sufficiently large when Compared to the frequency and voltage scaling Overhead.
2. Better if the target processor does not support out Of order execution.



$$\text{scenario I}: f_{\text{cpu}}^{\text{onchip}} = \frac{W^1_{\text{onchip}} + W^3_{\text{onchip}}}{D - \left(\frac{W^3_{\text{offchip}} + W^4_{\text{offchip}}}{f_{\text{mem}}}\right)}$$

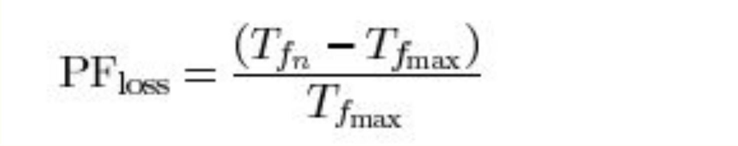
$$f_{\text{cpu}}^{\text{offchip}} = f_{\text{cpu}}^{\text{min}}$$

$$\text{scenario II}: f_{\text{cpu}}^{\text{onchip}} = f_{\text{cpu}}^{\text{offchip}} = \frac{W^1_{\text{onchip}} + W^3_{\text{onchip}}}{D - \left(\frac{W^2_{\text{offchip}} + W^4_{\text{offchip}}}{f_{\text{mem}}}\right)}$$

B. Performance degradation and energy saving

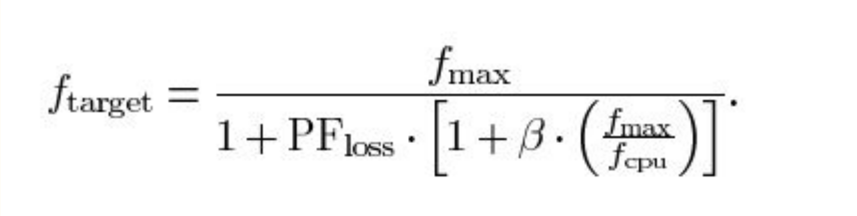
- Trade-off between performance and cpu frequency.

Performance loss :



$$\mathrm{PF}_{\mathrm{loss}} = \frac{(T_{f_n} - T_{f_{\max}})}{T_{f_{\max}}}$$

CPU frequency base on reducing the Performance loss :



$$f_{\mathrm{target}} = \frac{f_{\max}}{1 + \mathrm{PF}_{\mathrm{loss}} \cdot \left[1 + \beta \cdot \left(\frac{f_{\max}}{f_{\mathrm{cpu}}}\right)\right]}.$$

Beta represents the degree of

Energy saving.

IV. REGRESSION BASED FINE-GRAINED DVFS

Presented by Thalagala B. P. - 180631J

A. Calculating β With a Regression Equation

- Higher the accuracy of **β**, higher the optimality of the frequency for a given performance loss.
- **Calculating the exact β statically during compile time is difficult** due to the unpredictable dynamic behavior of microprocessors.
- β is heavily depend on 1) the **number of instructions being executed** and 2) the **number of external memory accesses.**
- Assuming that parameter β for the next time unit (t+1) is the same as that for the current time unit (t), it is calculated with the help of a regression equation.

$$f_{\mathrm{target}} = \frac{f_{\max}}{1 + \mathrm{PF}_{\mathrm{loss}} \cdot \left[1 + \beta \cdot \left(\frac{f_{\max}}{f_{\mathrm{cpu}}}\right)\right]}$$

$$\mathrm{CPI}^{\mathrm{avg}} = b(f) \cdot \mathrm{MPI}^{\mathrm{avg}} + c$$

$$\beta^{t} = \frac{\mathrm{CPI}^{\mathrm{avg},t}}{\mathrm{CPI}^{\mathrm{avg},t}_{\mathrm{onchip}}} - 1.$$

$$f^{t+1} = \frac{f_{\max}}{1 + \mathrm{PF}_{\mathrm{loss}} \cdot \left[1 + \beta^{t} \cdot \left(\frac{f_{\max}}{f^{t}}\right)\right]}$$

B. Prediction-Error Adjustment

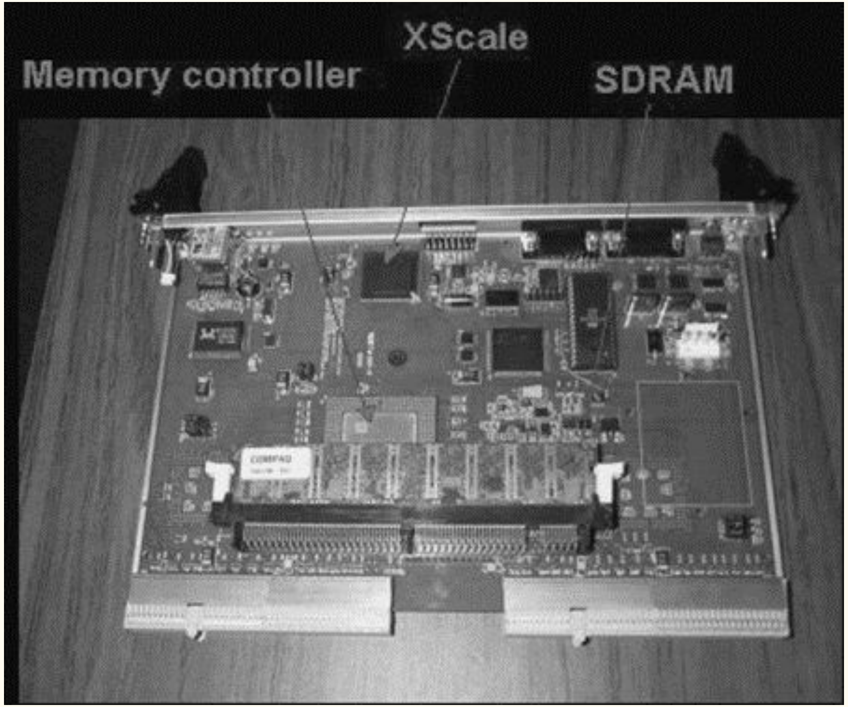
- In reality β can **vary significantly even within one time unit** depending on the characteristics of the target-application program.
- **Fluctuation of β is severe for memory-bound applications** and becomes worse when the time unit's length is also a variable.
- To facilitate this variable length in a time unit, a new term is introduced to **capture slack time ((expected - actual) execution time)** into the equation.

$$f^{t+1} = \frac{f_{\max}}{1 + \mathrm{PF}_{\mathrm{loss}} \cdot \left[1 + \left(\beta^t + \frac{S^t}{\mathrm{PF}_{\mathrm{loss}} \cdot T^t_{\mathrm{act}}}\right) \cdot \left(\frac{f_{\max}}{f^t}\right)\right]}$$

V.IMPLEMENTATION

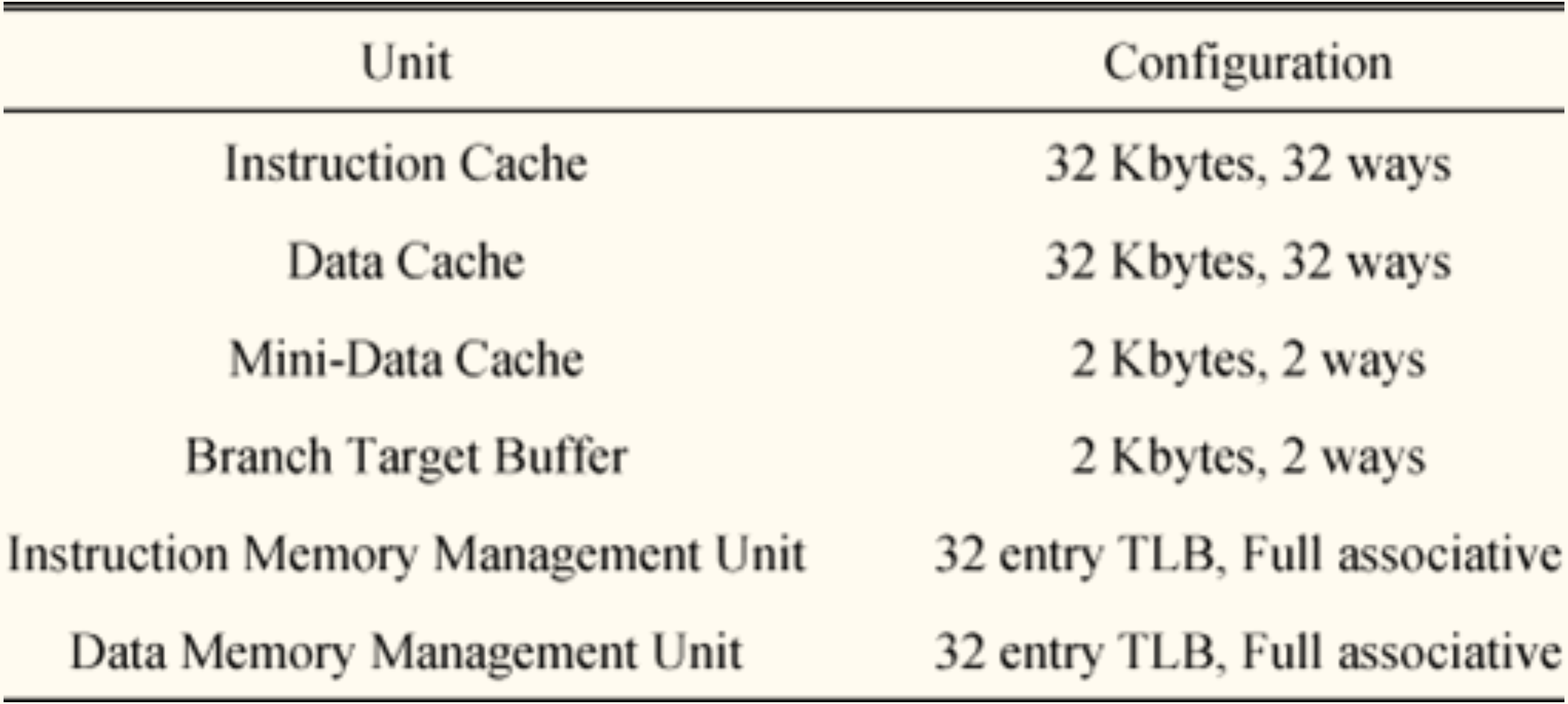
Presented by Rathnayake R.M.A.S - 180534N

Implementation - Testbed configuration



- Intel 80 200 Xscale microprocessor
- 128 MByte , 64 bit bus SDRAM
- PCI host bridge
- Flash memory, UART, DMA & IRQ controllers
- Xilinx VirtexE/VirtexII FPGA chip
- LCD, DSP, Ethernet, USB 2.0 PCMCIA modules
- Linux (v2.4.17)

Implementation - Intel 80 200 Xscale processor



Unit	Configuration
Instruction Cache	32 Kbytes, 32 ways
Data Cache	32 Kbytes, 32 ways
Mini-Data Cache	2 Kbytes, 2 ways
Branch Target Buffer	2 Kbytes, 2 ways
Instruction Memory Management Unit	32 entry TLB, Full associative
Data Memory Management Unit	32 entry TLB, Full associative

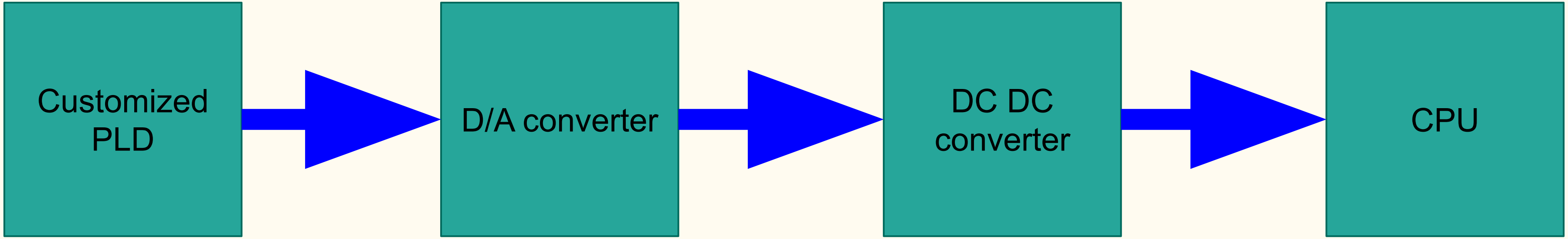
Implementation - CPU clock controller

PLL

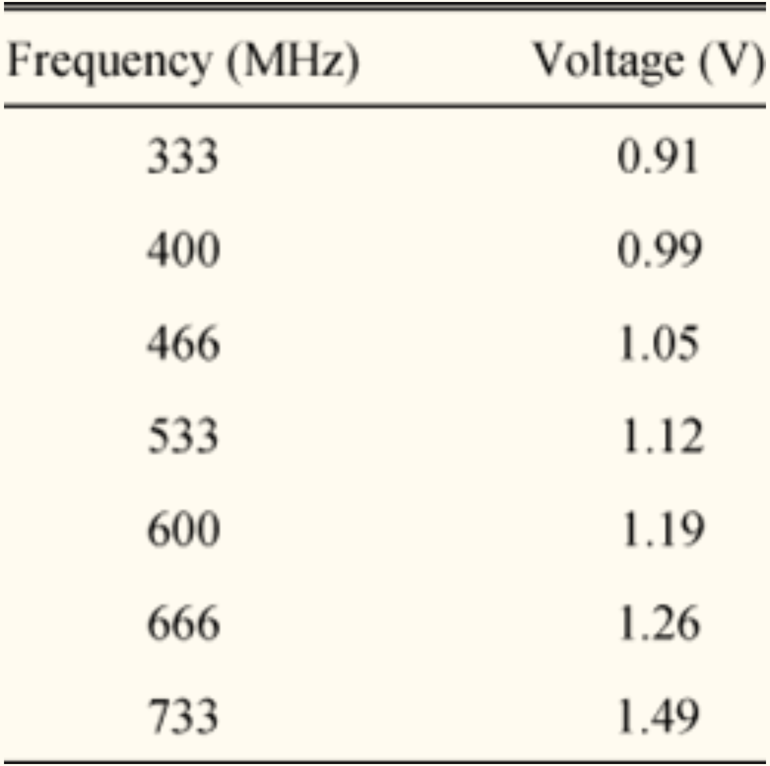
$$\text{bus speed} < \frac{\text{CPU clk}}{3}$$

```
for (clock=200; clock<=733; clock+=66)
```

Implementation - Variable voltage generator

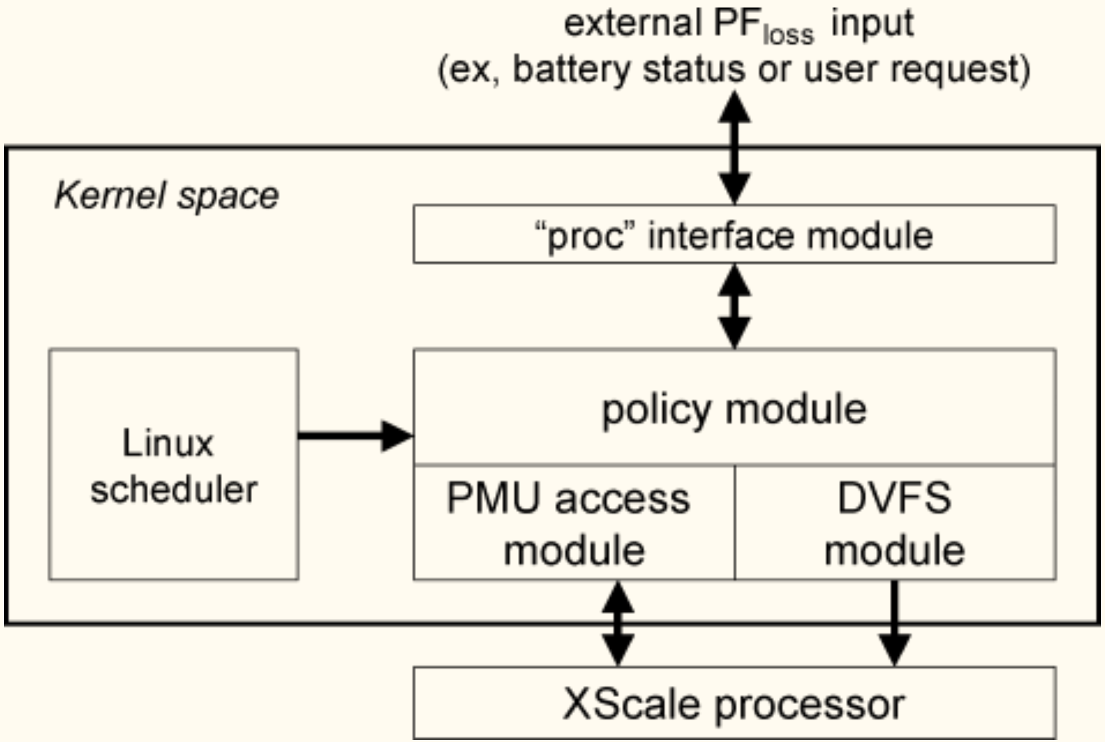


Implementation - Minimum voltage levels



Frequency (MHz)	Voltage (V)
333	0.91
400	0.99
466	1.05
533	1.12
600	1.19
666	1.26
733	1.49

Implementation - DVFS module and architecture



VI.EXPERIMENTAL RESULTS

Presented by Thanujaya M.G.S.- 180634V

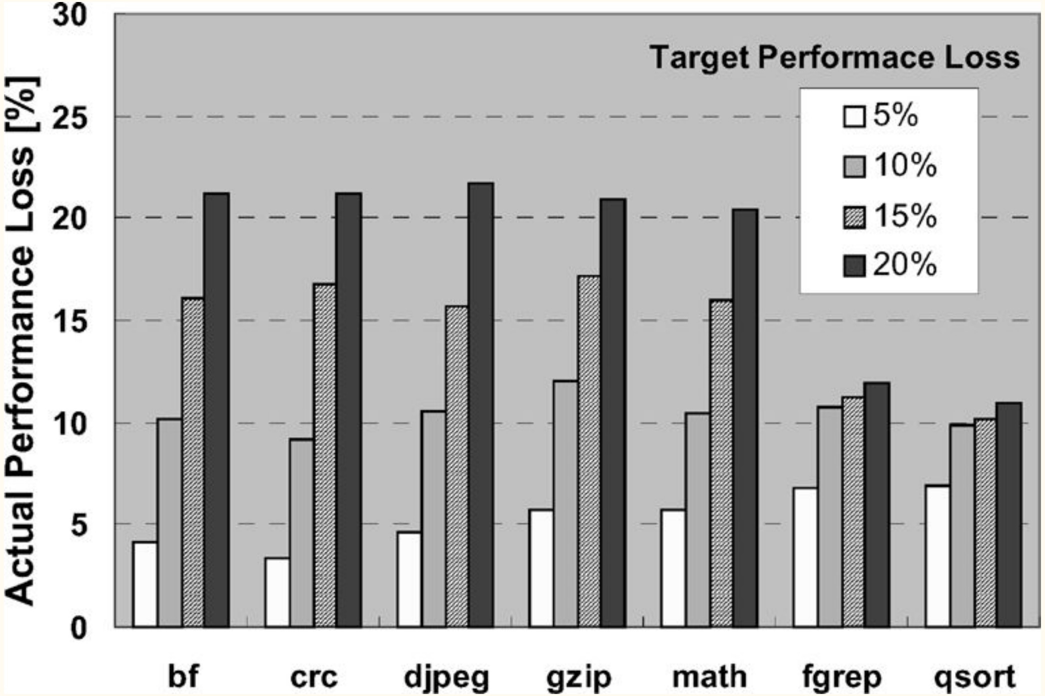
Test applications

- All the measurements are performed 10 times for each benchmark
- The **average performance loss** and **average energy saving values** are reported.
- Size of the window (N) =25
- N of 20~50 shows similar characteristics.

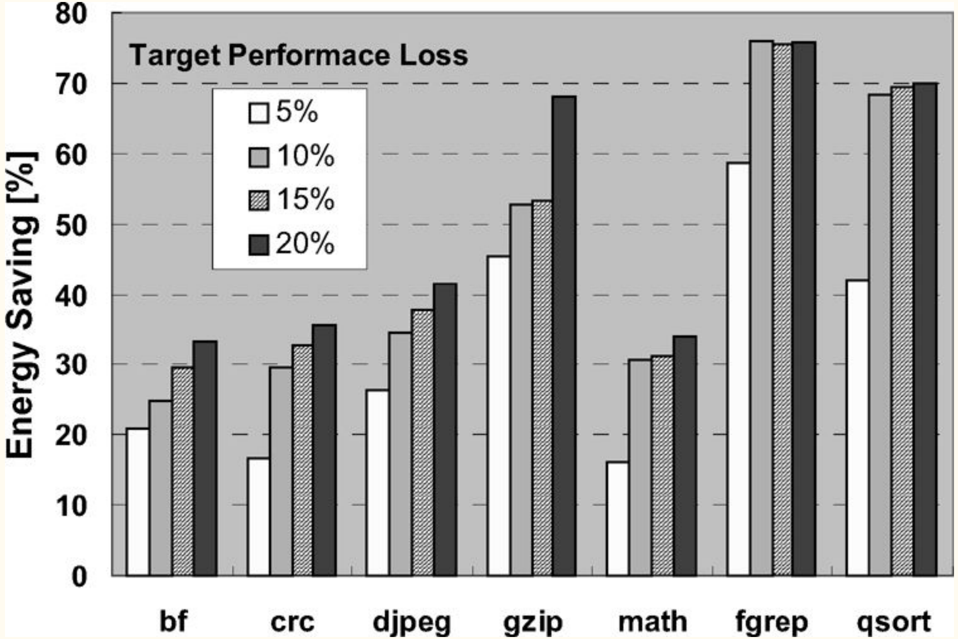
Benchmarks	Description
gzip	Compressing a given input
fgrep	Searching for a given pattern in the files residing in a directory
math	Floating-point calculations
bf(blowfish)	A symmetric block cipher with a variable length key from 32 to 448 bits
crc	32-bit cyclic redundancy check on a file
djpeg	Decoding a jpeg image file
qsort	Sorting a large array of string in ascending order

Summary of test applications

Performance loss with different target values

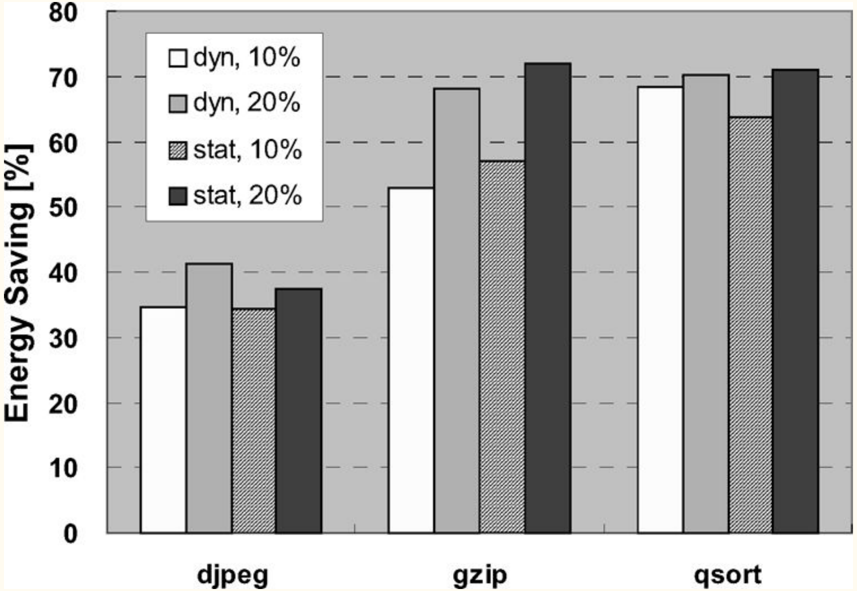


CPU energy saving



Comparison of dynamic versus static approach

Benchmarks	β_{avg}	f_{opt}(Calculated frequency) [MHz]		f_{app} (Applied frequency) [MHz]	
		10% PF_{loss}	20% PF_{loss}	10% PF_{loss}	20% PF_{loss}
djpej	0.49	637.9	564.7	666	600
gzip	5.26	450.8	325.5	466	333
qsort	7.82	389.5	265.2	400	333



Calculated CPU frequency for test application after profiling.

VII.CONCLUSION

Conclusion

- A regression based DVFS policy for finely tunable energy performance tradeoff
- The CPU voltage/frequency is scaled based on the ratio of the on-chip and off-chip latencies (β).
- β calculate using a regression equation, which is dynamically updated.
- For memory-bound programs more than 70% CPU energy saving with about 12% performance degradation.
- For CPU-bound programs, 15%60% energy saving with fine-tuned performance degradation, ranging 5% to 20%.