Power Management Using Online Learning

CSE-237A Lecture
Outline

- Design
- Implementation
- Experimental Results
- Analysis
Outline

- Design
- Implementation
- Experimental Results
- Analysis
DPM

Fundamentally a decision problem

Energy Savings/Performance Delay Tradeoff

Workload

Request

\( T_{idle} \)

Request

State

Active

Off

Active

Shutdown Cost

\( T_{active\rightarrow off} \)

\( T_{off} \)

\( T_{off\rightarrow active} \)

Power

\( P_{active} \)

\( P_{active\rightarrow off} \)

\( P_{off\rightarrow active} \)

Wakeup Cost

Time
Break Even Time

- Minimum idle time for amortizing the cost of component shutdown

\[ T_{off}P_{off} + T_{tr}P_{tr} = T_{idle}P_{active} \]

\[ T_{be} = T_{idle} = T_{tr} + T_{off} \]

Workload

\[ T_{idle} \]

State

\[ T_{active \rightarrow off} \]

\[ T_{off} \]

\[ T_{off \rightarrow active} \]

\[ T_{idle} < T_{be} \Rightarrow \text{No energy savings possible} \]
DPM Policies

Software entities that take DPM decisions

- **Timeout**: Karlin, Irani TECS’03
- **Predictive**: Srivastava, Hwang ICCAD’99
  - Heuristic: No performance guarantees
- **Stochastic**: DTMDP (Benini TCAD’00), TISMDP (Simunic TCAD’01)
  - Optimality for stationary workloads

**Observation:**

- No flexibility in terms of user perceived energy savings performance delay tradeoff
- Policies out perform each other under different workloads
Problem Formulation

- Take a set of existing DPM policies/experts
  - Perform dynamic selection at run time

DPM becomes a problem of: *characterization* and *expert selection*
DVFS + DPM (CPU)

- DVFS: problem of selection among multiple v-f settings
- Low v-f setting
  - Prolonged execution time
    - Extra leakage power and performance
  - Shorter idle periods
    - Impacts DPM

High v-f setting

Low v-f setting
Effectiveness of DVFS

- Depends on:
  - Impact on execution time
    - Task characteristics (CPU-intensiveness)
  - Extra leakage power consumption
    - Leakage characteristics of CPU

- Problem Formulation
  - “Select best suited v-f setting (expert) based on the characterization of the executing task and CPU leakage”
Modeling

- Characterization
  - DPM: idle period duration distribution
  - DVFS: task and CPU leakage

- Expert Selection
  - DPM: A state of the art DPM policy
  - DVFS: v-f setting

- Online Learning algorithm to solve this problem (Freund, JCSS’97)
  - Theoretical Guarantee on convergence to best suited expert
Online Learning for Horse Racing

Experts

Selects the best performing expert for investing his money

Expert manages money for the race

Evaluates performance of all experts for that race
Online Learning for Power Management

Experts (Working Set)

Selected expert manages power for the operative period

Selects the best performing expert for managing power

Evaluates performance of all the experts

Controller

Device

EXP y: Dormant Experts

EXP y: Operational Expert

EXP 1, EXP 2, EXP 3, ...., EXP n
Controller Algorithm

Parameters: $\beta \in [0,1]$

Initial weight vector for experts $w_i^1 \in [0,1]^N$
such that $\sum_{i=1}^{N} w_i^1 = 1$

Do for operative period $t = 1, 2, 3, \ldots$

1. Choose expert with highest probability factor in $r^t$
   \[ r^t = \frac{w^t}{\sum_{i=1}^{N} w_i^t} \]

2. Operational expert takes control of the device
3. Operative period ends -> evaluate performance of experts
4. Set the new weight vector to be:
   \[ w_i^{t+1} = w_i^t \beta_i^t \]

Loss factor: “un”suitability of an expert for the current workload
Performance bound on Controller

Controller converges to the best performing expert with successive operative periods

Let \( N \): experts in working set, \( T \): total number of operative periods

- If \( l^t_i \) is the loss incurred by expert \( i \):
  \[
  \sum_{i=1}^{N} r^t_i l^t_i = r^t . l^t
  \]
  Loss incurred by controller

- Goal: minimize net loss \((L_G - \min_i L_i)\)
  \[
  L_G = \sum_{t=1}^{T} r^t . l^t \\
  L_i = \sum_{t=1}^{T} l^t_i
  \]
  Total loss incurred by controller, Total loss incurred by best expert

- Average net loss per period decreases at the rate of \( O\left(\sqrt{\frac{(\ln N)}{T}}\right) \)
Outline

- Design
- Implementation
- Experimental Results
- Analysis
DPM

- Operative period == Idle period
  - Loss calculation ($l_i^t$): performance baselined against the oracle policy

\[
l_{ie}^t = 1 - \frac{\text{time policy ‘i’ sleeps}}{\text{time oracle sleeps}}
\]

- $l_i^t$ broken into energy and performance components:
  \[
l_i^t = \alpha l_{ie}^t + (1 - \alpha) l_{ip}^t
\]
- “$\alpha$” factor controls the energy savings/performance delay (e/p) tradeoff
Weight Vector

- Weight vector is updated every idle period
  - Penalize the under-performing experts
- Converges to the best performing expert
- Maps suitability of experts to the current workload

\[ \text{Weight vector} = \text{Workload characteristics} \]
CPU (DVFS+DPM)

Energy Model

- Power = Dynamic($V^2f$) + Leakage($VI_{leakage}$)
  \[ P = D + L \]
- Define $P_n$, $V_n$, $f_n$ as normalized quantities
  \[ P_n = \frac{P}{P_{max}} \]
- Define $\rho$ as leakage percentage
  \[ \rho = \frac{L_{max}}{P_{max}} \]
  \[ P_n = (1 - \rho)V_n^2f_n + \rho V_n \]
  \[ E_n = ((1 - \rho)V_n^2f_n + (\rho V_n)T_n) \]
Execution Time

- $T = T_{cpu} + T_{stall}$
  - $T_{cpu}$: frequency dependent
  - $T_{stall}$: frequency invariant

$$\mu = \frac{T_{cpu}}{T}$$

- High $\mu$ => High CPU intensiveness:
  - High impact on $T_n$
Importance of $\mu$

Defined three tasks:
1. burn_loop ($\mu \approx 1$)
2. mem ($\mu \approx 0$)
3. combo ($\mu \approx 0.5$)

\[ E_n = ((1 - \rho)V_n^2 f_n + \rho V_n).T_n \]

Intel PXA27x Processor

$\rho = 27\%$
Impact of $\rho$

$$E_n = ((1 - \rho)V_n^2 f_n + \rho V_n).T_n$$

mem

burn_loop

\begin{align*}
\text{mem} &;\quad \text{burn\_loop} \\
\text{40\%} &;\quad \text{40\%} \\
\text{60\%} &;\quad \text{60\%} \\
\text{80\%} &;\quad \text{80\%} \\
\text{100\%} &;\quad \text{100\%} \\
\text{40\%} &;\quad \text{110\%} \\
\text{60\%} &;\quad \text{110\%} \\
\text{80\%} &;\quad \text{110\%} \\
\text{100\%} &;\quad \text{110\%} \\
\end{align*}
Controller

- Operative period == scheduler quantum
  - Suitability of v-f setting expert depends on:
    - Task characteristics (µ)
    - CPU leakage characteristics (ρ)

- Quantify task characteristics (µ)
  - **CPI Stack**
    \[ \text{CPI}_{\text{avg}} = \text{CPI}_{\text{base}} + \text{CPI}_{\text{stall}} \]
  - Estimate \( \mu = \frac{\text{CPI}_{\text{base}}}{\text{CPI}_{\text{avg}}} \)
Controller (contd.)

- Dynamic task characterization
  - Dynamically estimate $\mu$ for every scheduler quantum
- $\mu$-mapper
  - Incorporates leakage characteristics ($\rho$)
  - Maps $\mu$ to the best suited v-f setting expert
\( \mu \)-Mapper

Eg: Expert1-5=\{100,200,300,400,500\}MHz

Loss calculation -> Comparison of \( \mu \) and \( \mu \)-means of experts

Energy Loss = \( (0.9 - 0.7) = 0.2 \)  
Performance Loss = 0
ρ = 50%
Weight Vector

- Characterizes suitability of v-f experts for executing task
- Multi-tasking systems?

Possible for task with differing characteristics to be runnable

**Solution:** store weight vector as a task level structure
Implementation

- Initialize weight vector on task basis
- To check if CPU idle
- To update weight vector

Linux Kernel

- Task Creation
- Context Switch
- Scheduler Tick

Controller

- User Preference
- PM decisions

CPU

User

/proc

Linux process Manager
Outline

○ Design
○ Implementation
○ Experimental Results
○ Analysis
Experiments

- Devices
  - Hard Disk Drive (HDD): HP 2200A
  - CPU: Intel PXA27x processor

- Device characteristics
  - Power states consumption and transition times

<table>
<thead>
<tr>
<th>Device</th>
<th>$P_{on}$</th>
<th>$P_{sleep}$</th>
<th>$P_{tr}$</th>
<th>$T_{tr}$</th>
<th>$T_{be}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDD</td>
<td>1.6W</td>
<td>0.4W</td>
<td>2.4W</td>
<td>2.5s</td>
<td>4.2s</td>
</tr>
<tr>
<td>CPU</td>
<td>747mW</td>
<td>1.7mW</td>
<td>747mW</td>
<td>12ms</td>
<td>12ms</td>
</tr>
</tbody>
</table>
DPM (HDD)

- Varying workload characteristics

<table>
<thead>
<tr>
<th>Device</th>
<th>Trace Name</th>
<th>$t_{RI}$</th>
<th>$\sigma_{t_{RI}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDD</td>
<td>HP-1 Trace</td>
<td>20.5</td>
<td>29</td>
</tr>
<tr>
<td></td>
<td>HP-2 Trace</td>
<td>5.9</td>
<td>8.4</td>
</tr>
<tr>
<td></td>
<td>HP-3 Trace</td>
<td>17.2</td>
<td>2</td>
</tr>
</tbody>
</table>

$t_{RI}$ : Average Request Inter-arrival Time (in sec)

- Working set of different classes of policies

<table>
<thead>
<tr>
<th>Expert</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed Timeout</td>
<td>Timeout = $7T_{be}$</td>
</tr>
<tr>
<td>Adaptive Timeout (Douglis, USENIX’95)</td>
<td>Initial timeout = $7T_{be}$; Adjustment = $+1T_{be}$/$-1T_{be}$</td>
</tr>
<tr>
<td>Exponential Predictive (Hwang, ICCAD’97)</td>
<td>$I_{n+1} = \alpha I_n + (1 - \alpha)I_n$ with $\alpha = 0.5$</td>
</tr>
<tr>
<td>TISMDP (Simunic, TCAD’01)</td>
<td>Optimized for delay constraint of 3.4% on HP-1 trace</td>
</tr>
</tbody>
</table>
## Results: Perf Delay/Energy Saving

### With Individual Experts

<table>
<thead>
<tr>
<th>Policy</th>
<th>HP1 Trace</th>
<th>HP2 Trace</th>
<th>HP3 Trace</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>%delay</td>
<td>%energy</td>
<td>%delay</td>
</tr>
<tr>
<td>Oracle</td>
<td>0</td>
<td>68.17</td>
<td>0</td>
</tr>
<tr>
<td>Timeout</td>
<td>4.2</td>
<td>49.9</td>
<td>4.4</td>
</tr>
<tr>
<td>Ad Timeout</td>
<td>7.7</td>
<td>66.3</td>
<td>8.7</td>
</tr>
<tr>
<td>TISMDP</td>
<td>3.4</td>
<td>44.8</td>
<td>2.26</td>
</tr>
<tr>
<td>Predictive</td>
<td>8</td>
<td>66.6</td>
<td>9.2</td>
</tr>
</tbody>
</table>

### With Controller

<table>
<thead>
<tr>
<th>Preference</th>
<th>HP-1 Trace</th>
<th>HP-2 Trace</th>
<th>HP-3 Trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>Least Delay</td>
<td>%delay</td>
<td>%energy</td>
<td>%delay</td>
</tr>
<tr>
<td>Low delay</td>
<td>3.5</td>
<td>45</td>
<td>2.61</td>
</tr>
<tr>
<td>High energy</td>
<td>6.13</td>
<td>60.64</td>
<td>5.86</td>
</tr>
<tr>
<td>savings</td>
<td>7.68</td>
<td>65.5</td>
<td>8.59</td>
</tr>
</tbody>
</table>
Results: Frequency of Selection

For HP-3 Trace

Results converge to TISMDP for low $\alpha$.

Predictive & Ad Timeout perform best for energy savings.
Selection with Fixed Timeout Experts

For HP-1 Trace

Experts:
T_{be} to 180s

Excellent performance with simple fixed timeout experts
CPU (DVFS + DPM)

- Working set:
  - 10 DPM experts (used fixed timeout experts)
    - $T_{be}$ to $9.0 T_{be}$
  - Four v-f setting experts

- Workloads:
  - *Idle dominated*
    - Editor
    - Web Surfing
  - *Computationally intensive*
    - qsort
    - djpeg
    - blowfish
    - dgzip
    - burn_loop

<table>
<thead>
<tr>
<th>Freq (MHz)</th>
<th>Voltage (V)</th>
</tr>
</thead>
<tbody>
<tr>
<td>208</td>
<td>1.2</td>
</tr>
<tr>
<td>312</td>
<td>1.3</td>
</tr>
<tr>
<td>416</td>
<td>1.4</td>
</tr>
<tr>
<td>520</td>
<td>1.5</td>
</tr>
</tbody>
</table>
## Results: Computationally Intensive

<table>
<thead>
<tr>
<th>Bench.</th>
<th>Low perf delay --------&gt; Higher energy savings</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>%delay</td>
</tr>
<tr>
<td>qsort</td>
<td>6</td>
</tr>
<tr>
<td>djpeg</td>
<td>7</td>
</tr>
<tr>
<td>dgzip</td>
<td>15</td>
</tr>
<tr>
<td>bf</td>
<td>6</td>
</tr>
<tr>
<td>burn_loop</td>
<td>0</td>
</tr>
</tbody>
</table>

- **Avg ~7% perf delay**
- **Avg ~37% energy savings**

<table>
<thead>
<tr>
<th>Bench.</th>
<th>208MHz/1.2V</th>
<th>%delay</th>
<th>%energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>qsort</td>
<td>56</td>
<td>48</td>
<td></td>
</tr>
<tr>
<td>djpeg</td>
<td>34</td>
<td>54</td>
<td></td>
</tr>
<tr>
<td>dgzip</td>
<td>33</td>
<td>54</td>
<td></td>
</tr>
<tr>
<td>bf</td>
<td>40</td>
<td>51</td>
<td></td>
</tr>
<tr>
<td>burn_loop</td>
<td>150</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

- **Within 7% of max energy savings**
- **~35% lower avg perf delay**
Result: Frequency of Selection

For qsort

Identifies both CPU-intensive and mem intensive phases correctly

Average μ:
- 0.4 for CPU-intensive
- 0.75 for mem intensive

Frequency of Selection:
- 25% CPU intensive
- 75% mem intensive

Time:
- 208MHz
- 312MHz
- 416MHz
- 520MHz

Energy savings:
- Lower Perf
- Delay
- Higher energy savings
Leakage Awareness

- Derived a $\mu$-mapper for 50% leakage

burn_loop (high energy savings preference)

416MHz

520MHz
Outline

- Design
- Implementation
- Experimental Results
- Analysis
Energy Savings with DVFS

\[ P_R = P_{c_1} - P_{c_2} \]

Reduction in CPU power

\[ P_E = (P_{c_2} - P_{c_{idle}}) + (P_d - P_{d_{idle}}) \]

Extra system power

\[ E_{DVFS} = P_R t_1 - P_E t_{delay} \]
Effectiveness of DVFS

\[ E_{DVFS} = P_R t_1 - P_E t_{delay} \]
\[ = E_R - E_E \]

- For energy savings
  - \( E_R > E_E \)
- Factors in modern systems affecting this equation:
  - Performance delay (\( t_{delay} \))
  - Idle CPU power consumption (\( P_E \))
  - Power consumption of other devices (\( P_E \))
Performance Delay

- Lower $t_{\text{delay}} \Rightarrow$ higher energy savings
  - Depends on memory/CPU intensiveness
- Experiments with SPEC CPU2000
  - $mcf$: highly memory intensive
    - Expect low $t_{\text{delay}}$
  - $sixtrack$: highly cache/CPU intensive
    - Expect high $t_{\text{delay}}$
- Two state of the art processors
  - AMD quad core Opteron
    - On die memory controller (2.6GHz), DDR3
  - Intel quad core Xeon
    - Off chip memory controller (1.3GHz), DDR2
Performance Delay

Due to on die memory controller and fast DDR3 memory

Due to slower memory controller and memory

mcf much closer to best case on Xeon

mcf much closer to worst case on AMD
Idle CPU power consumption

- Low power idle CPU states common now
  - C1 state used be default
    - Zero dynamic power consumption
- Support for deeper C-states appearing
  - C6 on Nehalem
    - Zero dynamic+leakage power

\[ P_E = (P_{c2} - P_{c_{idle}}) + (P_d - P_{d_{idle}}) \]

- Higher extra CPU power consumption for modern CPUs
  - Lower DVFS benefits
Device power consumption

- DVFS makes other devices consume power for longer time ($t_{\text{delay}}$)
  - Memory (4GB DDR3)
    - Idle -> 5W
    - Active -> 10W

\[ P_E = (P_{c2} - P_{c_{idle}}) + (P_d - P_{d_{idle}}) \]

- Higher extra device power consumption
  - Lower DVFS benefits for memory intensive benchmarks
Evaluation Setup

- Assume a simple static-DVFS policy
  - AMD Opteron (four v-f settings):
    - 1.25V/2.6GHz, 1.15V/1.9GHz, 1.05V/1.4GHz, 0.9V/0.8GHz
- Compare against a base system with no DVFS and three simple idle PM policies:

<table>
<thead>
<tr>
<th>Policy</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PM-1</td>
<td>switch CPU to ACPI state C1 (remove clock supply) and move to lowest voltage setting</td>
</tr>
<tr>
<td>PM-2</td>
<td>switch CPU to ACPI state C6 (remove power)</td>
</tr>
<tr>
<td>PM-3</td>
<td>switch CPU to ACPI state C6 and switch the memory to self-refresh mode</td>
</tr>
</tbody>
</table>
Methodology

- Run SPEC CPU2000 benchmarks at all v-f settings
  - Estimate savings baselined against system with PM-(1:3) policies
  - \( \% E_{savings_{PM-i}} = \frac{E_{DVFSf}}{E_{PM-i}} \)
    - where \( i = 1, 2, 3 \)
  - \( E_{PM-i} \) varies based on the policy
  - DVFS beneficial if:
    - \( \% E_{savings_{PM-i}} > 0 \)
## Results

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Freq</th>
<th>%delay</th>
<th>%Energy&lt;sub&gt;savingsPM-i&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>PM-1</td>
</tr>
<tr>
<td>mcf</td>
<td>1.9</td>
<td>29</td>
<td>5.2</td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td>63</td>
<td>8.1</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>163</td>
<td>8.1</td>
</tr>
<tr>
<td>bzip2</td>
<td>1.9</td>
<td>37</td>
<td>4.7</td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td>86</td>
<td>7.4</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>223</td>
<td>7.8</td>
</tr>
<tr>
<td>art</td>
<td>1.9</td>
<td>32</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td>76</td>
<td>7.3</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>202</td>
<td>8</td>
</tr>
<tr>
<td>sixtrack</td>
<td>1.9</td>
<td>37</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td>86</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>227</td>
<td>7</td>
</tr>
</tbody>
</table>
## Results

- Max Avg ~7% savings
- High perf delay

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Freq</th>
<th>%delay</th>
<th>%Energy\textsubscript{savingsPM-1}</th>
<th>PM-1</th>
<th>PM-2</th>
<th>PM-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>mcf</td>
<td>1.9</td>
<td>29</td>
<td>5.2</td>
<td>0.7</td>
<td>-0.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td>63</td>
<td>8.1</td>
<td>0.1</td>
<td>-2.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>163</td>
<td>8.1</td>
<td>-6.3</td>
<td>-10.7</td>
<td></td>
</tr>
<tr>
<td>bzip2</td>
<td>1.9</td>
<td>37</td>
<td>4.7</td>
<td>-0.6</td>
<td>-2.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td>86</td>
<td>7.4</td>
<td>-2.4</td>
<td>-5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>223</td>
<td>7.8</td>
<td>-9.0</td>
<td>-14</td>
<td></td>
</tr>
<tr>
<td>art</td>
<td>1.9</td>
<td>32</td>
<td>6</td>
<td>1</td>
<td>-0.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td>76</td>
<td>7.3</td>
<td>-1.7</td>
<td>-4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>202</td>
<td>8</td>
<td>-8</td>
<td>-13</td>
<td></td>
</tr>
<tr>
<td>sixtrack</td>
<td>1.9</td>
<td>37</td>
<td>5</td>
<td>-0.5</td>
<td>-2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td>86</td>
<td>6</td>
<td>-4.3</td>
<td>-7.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>227</td>
<td>7</td>
<td>-11</td>
<td>-16.1</td>
<td></td>
</tr>
</tbody>
</table>
## Results

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Freq</th>
<th>%delay</th>
<th>%Energy&lt;sub&gt;savingsPM-i&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>PM-1</td>
</tr>
<tr>
<td>mcf</td>
<td>1.9</td>
<td>29</td>
<td>5.2</td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td>63</td>
<td>8.1</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>163</td>
<td>8.1</td>
</tr>
<tr>
<td>bzip2</td>
<td>1.9</td>
<td>37</td>
<td>4.7</td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td>86</td>
<td>7.4</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>223</td>
<td>7.8</td>
</tr>
<tr>
<td>art</td>
<td>1.9</td>
<td>32</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td>76</td>
<td>7.3</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>202</td>
<td>8</td>
</tr>
<tr>
<td>sixtrack</td>
<td>1.9</td>
<td>37</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td>86</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>227</td>
<td>7</td>
</tr>
</tbody>
</table>

- Lowest v-f setting not useful
- Avg 7% savings
- Avg 200% delay
### Results

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Freq</th>
<th>%delay</th>
<th>%Energy_{savings}^{PM-i}</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>PM-1</td>
</tr>
<tr>
<td>mcf</td>
<td>1.9</td>
<td>29</td>
<td>5.2</td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td>63</td>
<td>8.1</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>163</td>
<td>8.1</td>
</tr>
<tr>
<td>bzip2</td>
<td>1.9</td>
<td>37</td>
<td>4.7</td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td>86</td>
<td>7.4</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>223</td>
<td>7.8</td>
</tr>
<tr>
<td>art</td>
<td>1.9</td>
<td>32</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td>76</td>
<td>7.3</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>202</td>
<td>8</td>
</tr>
<tr>
<td>sixtrack</td>
<td>1.9</td>
<td>37</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td>86</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>227</td>
<td>7</td>
</tr>
</tbody>
</table>

- DVFS energy inefficient
- Lower system idle power consumption
Conclusion

- Simple power management policies provide better energy performance tradeoffs
- Lower v-f setting offer worse e/p tradeoffs due to high performance delay
- DVFS still useful for:
  - Power reduction: thermal management
  - Systems with simpler memory controllers and low power system components