# Dynamic Voltage Scheduling Using Adaptive Filtering of Workload Traces

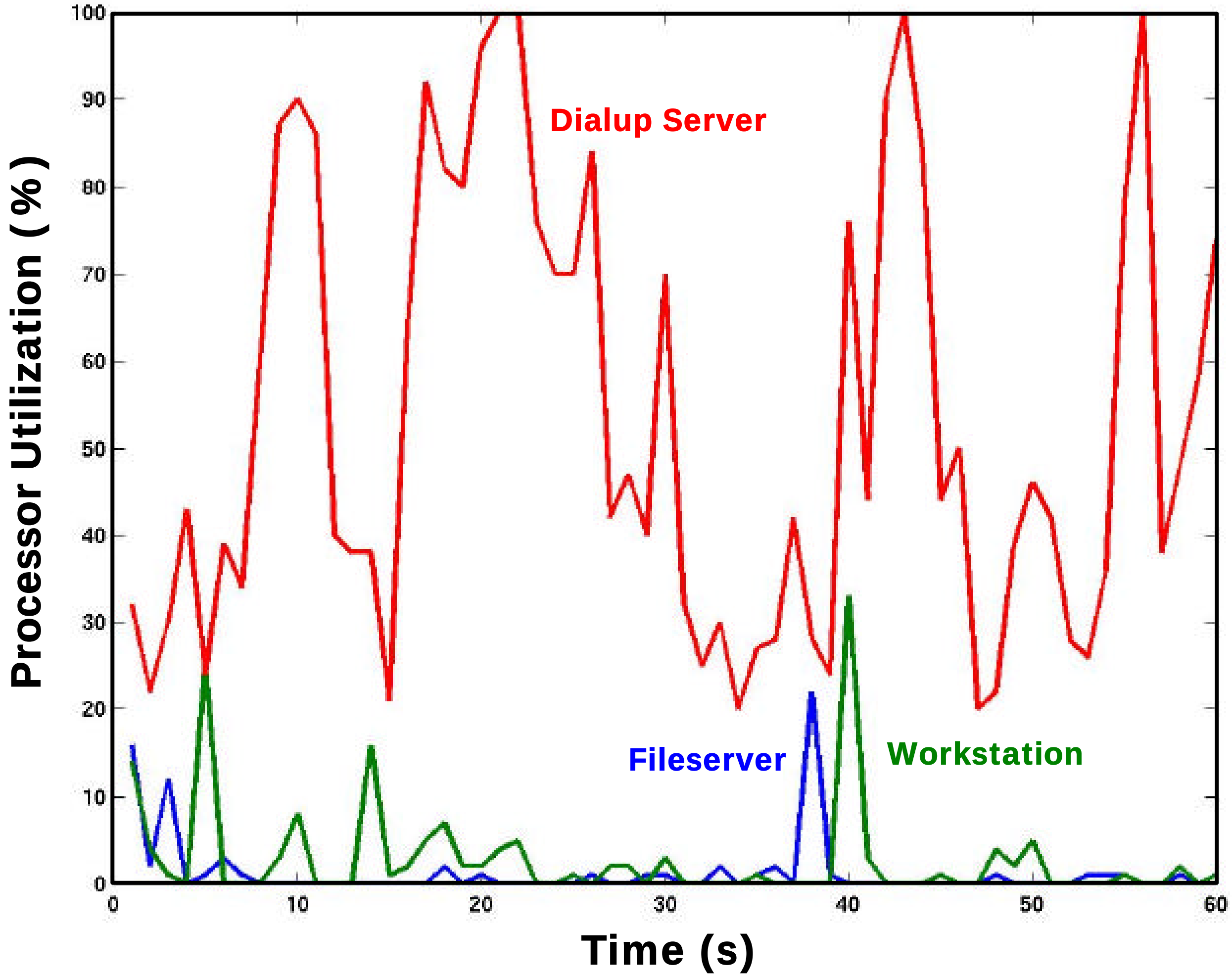
**Amit Sinha and Anantha Chandrakasan**

*Massachusetts Institute of Technology*

# Overview

- Introduction
  - Typical Workload Profile
  - DVS Basics
- Energy Workload Models
- Workload Prediction
  - Markov Processes
  - Various Algorithms
- Energy Performance Tradeoffs
- Results and Conclusions

# Typical Processor Workload Profiles



# Dynamic Voltage Scaling

**Fixed Power Supply**

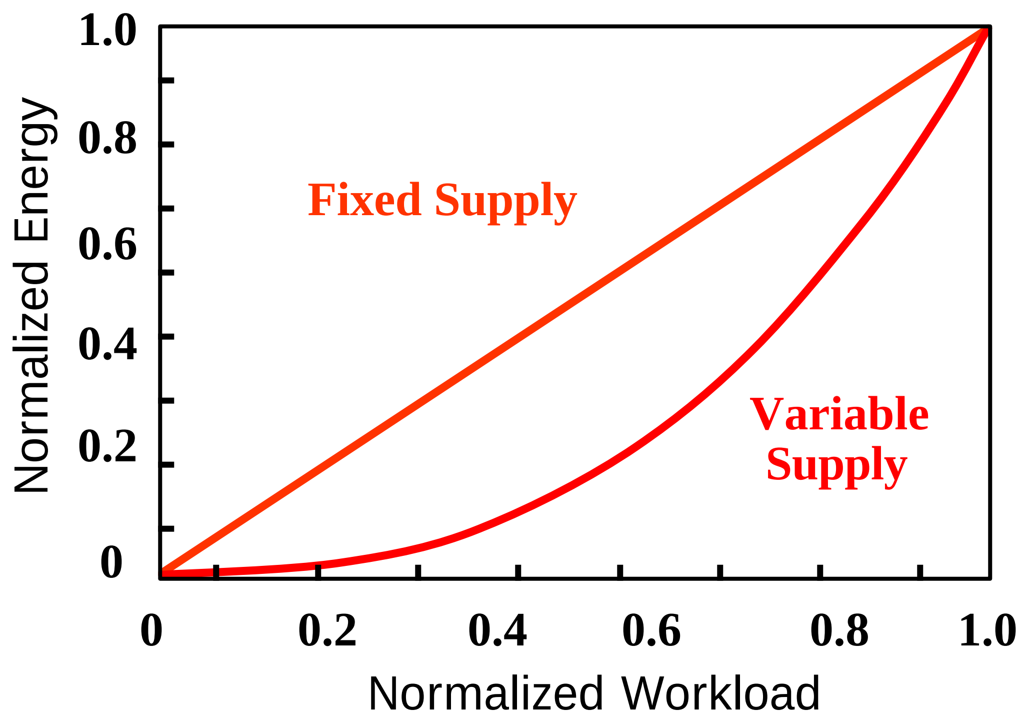
ACTIVE | IDLE

$E_{FIXED} = \frac{1}{2} C V_{DD}^2$

**Variable Power Supply**

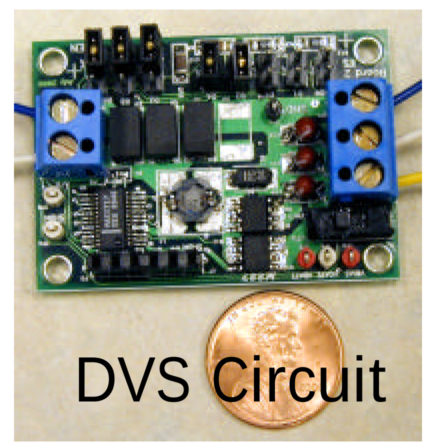
ACTIVE

$E_{VARIABLE} = \frac{1}{2} C (V_{DD}/2)^2 = E_{FIXED} / 4$



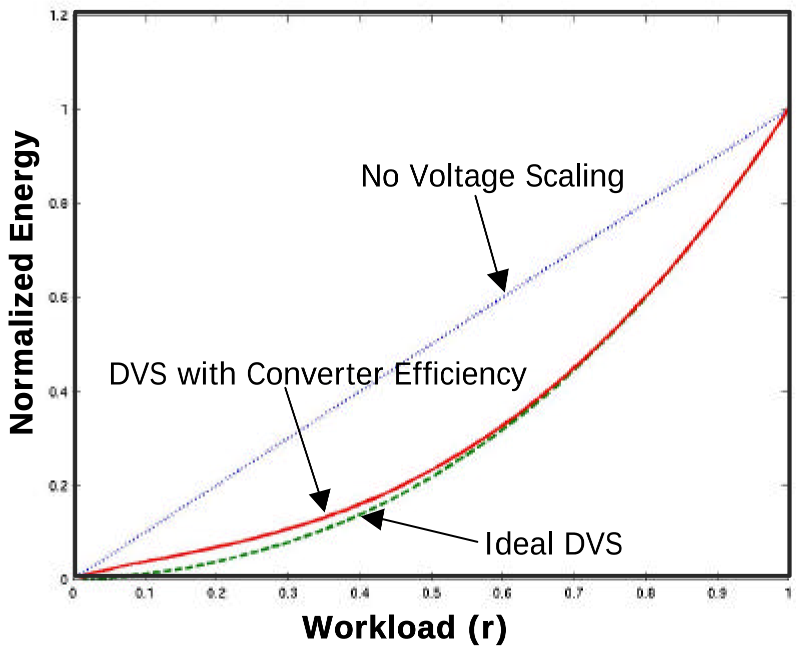
# Enabling Technology

- Variable frequency processors available
  - Transmeta's Crusoe
    - LongRun Technology
  - AMD K6-2+
    - PowerNOW!
  - Mobile Pentium III
    - SpeedStep
- StrongARM SA-1100
  - 59MHz – 206MHz (0.8V – 1.5V)

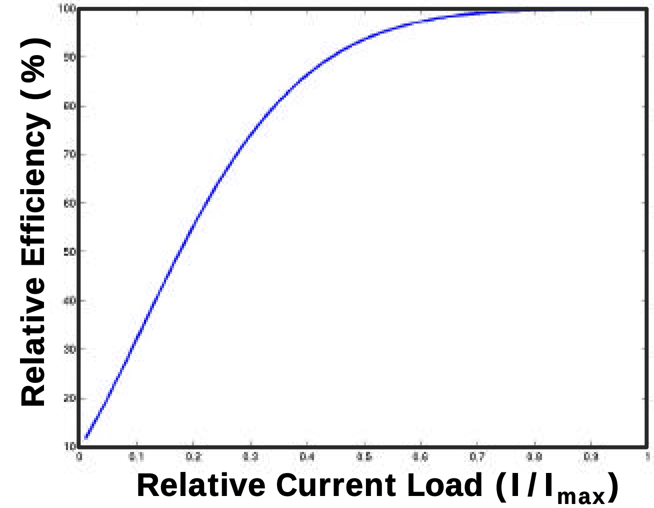



# Energy Workload Model

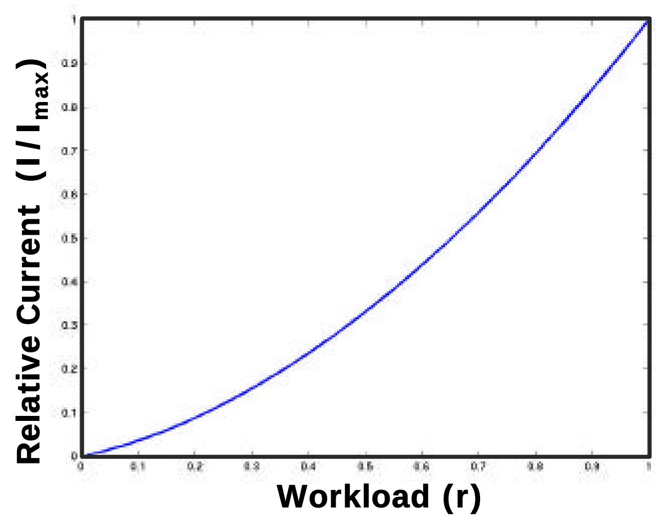
**Energy vs. Workload**




**DC / DC Efficiency**




**Current vs. Workload**



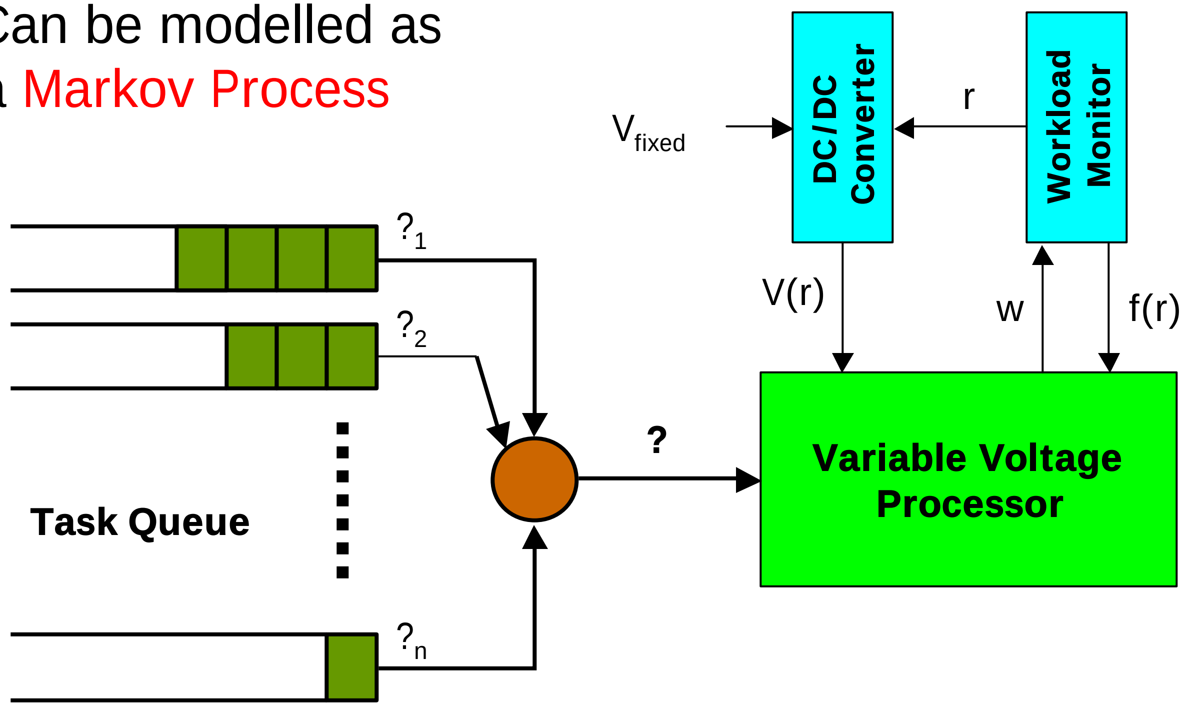

**[Gutnik97]**

$$E(r) = CV_0^2 T_s f_{ref} r \left[ \frac{V_t}{V_0} + \frac{r}{2} + \sqrt{r\frac{V_t}{V_0} + \left(\frac{r}{2}\right)^2} \right]^2$$

$$I(r) = I_{ref} r \frac{V_0}{V_{ref}} \left[ \frac{V_t}{V_0} + \frac{r}{2} + \sqrt{r\frac{V_t}{V_0} + \left(\frac{r}{2}\right)^2} \right]$$

# Workload Prediction

Can be modelled as
a Markov Process




- How to predict workload, w?
- How frequently processing rate, f(r), be updated

# Prediction Algorithms

**Predicted Workload**

$$w_p[n+1] = \sum_{k=0}^{N-1} h_n[k]\, w[n-k]$$

**Previous Workloads**

## Moving Average Workload (MAW)

$$h_n[k] = \frac{1}{N} \; \forall n, k$$

- Simplest
- Peformance degradation with fast loads

## Exp. Weighted Average (EWA)

$$h_n[k] = a^{-k}$$

- Lower significance of older data
- Event predictition context [Hwang97]

## Expected Workload State (EWS)

$$w[n+1] = \mathrm{E}\{w[n+1]\} = \sum_{j=0}^{L} w_j \, p_{ij}$$

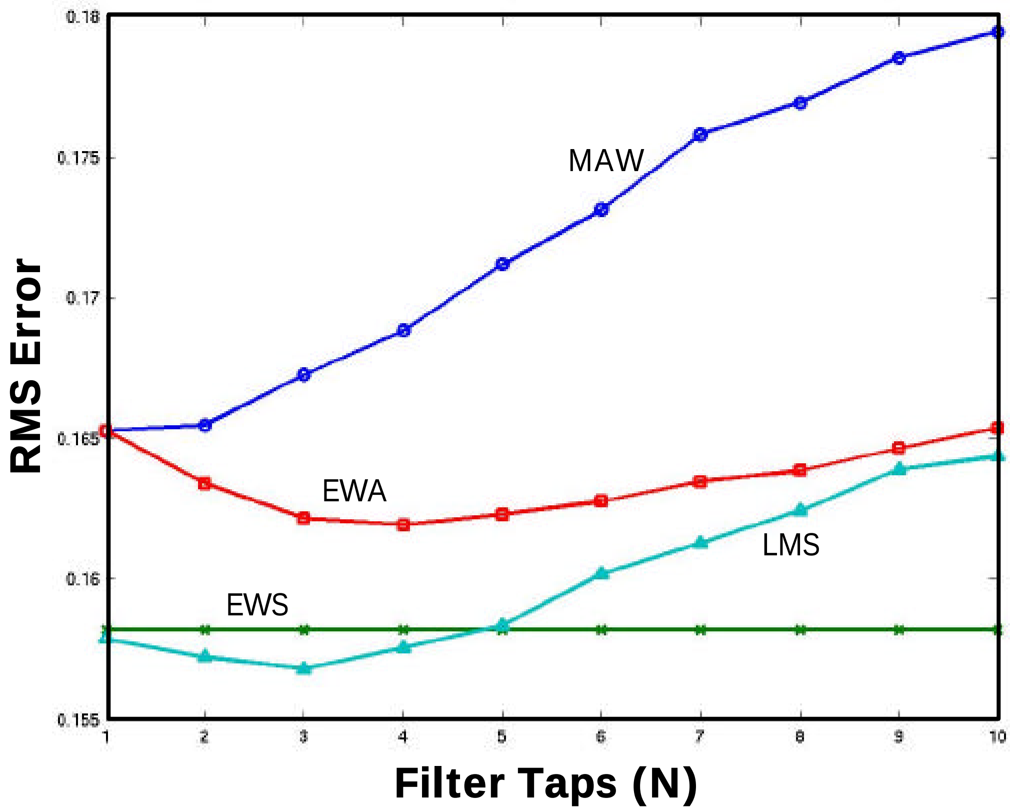
- Probabilistic fomulation
- Transition matrix updated every slot

## Least Mean Square (LMS)

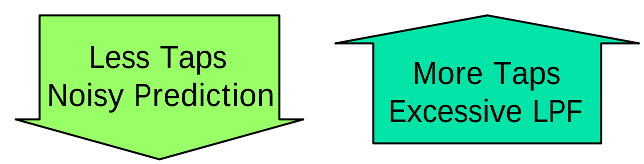
$$h_{n+1}[k] = h_n[k] + m w_e[n] w[n-k]$$

- Adaptive filter, self-adjusting
- Convergence issues

# Prediction Performance

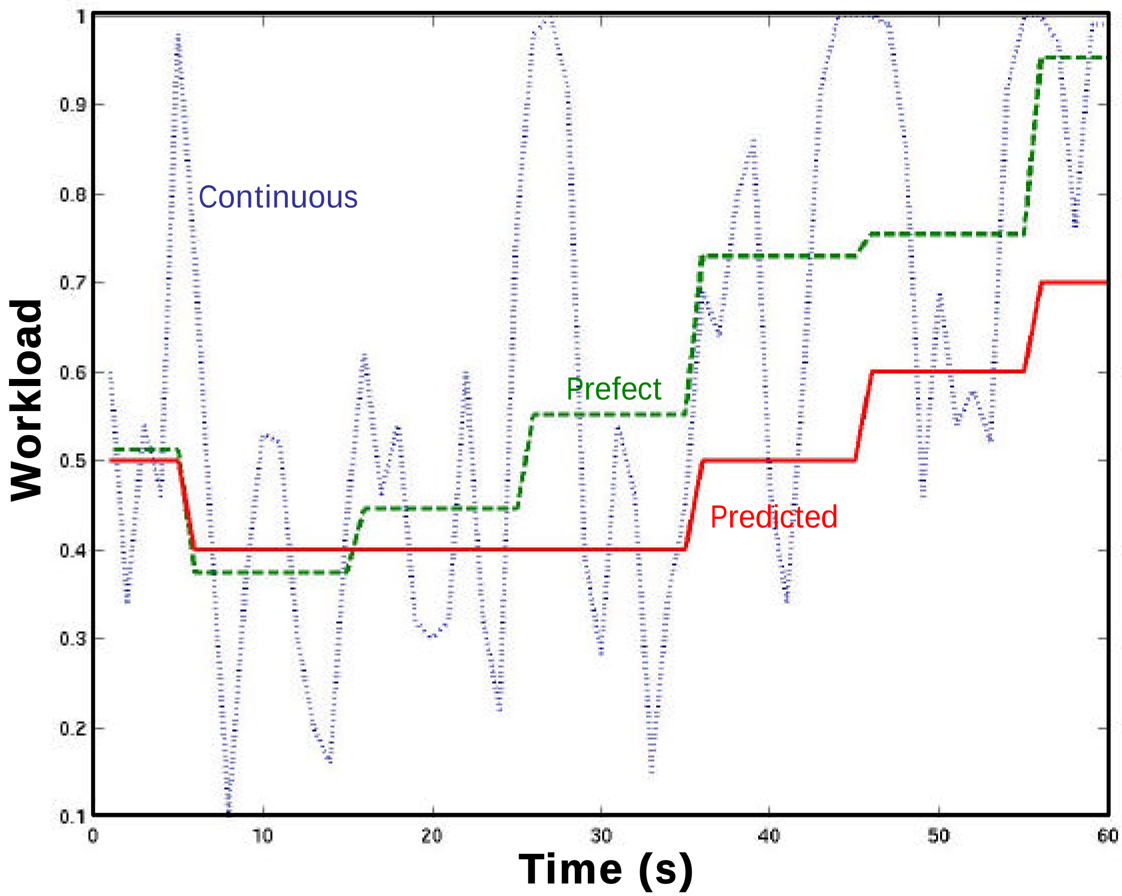



- Averaged over different processors and times
- 1 sec update rate
- 1 hour processor utilization snapshots




- Best prediction with LMS and about 3 taps

# LMS Tracking of Workload

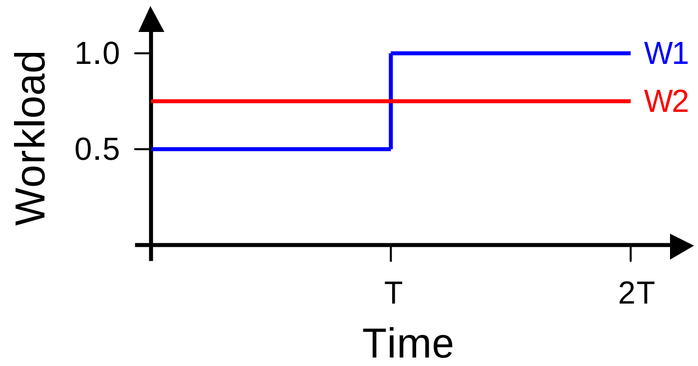


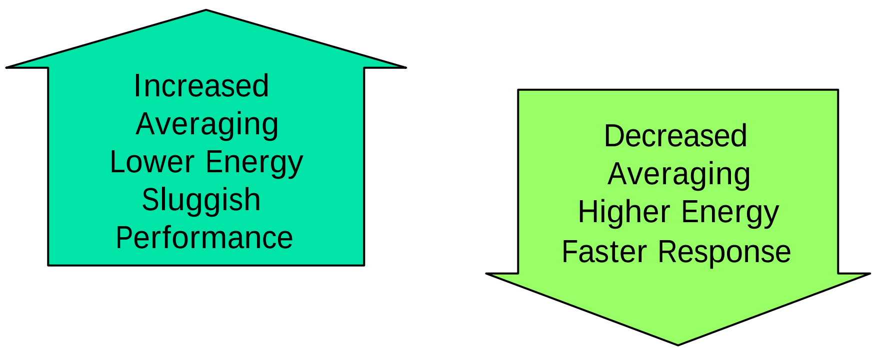

N = 3
T = 10
Levels = 10
μ = 0.1

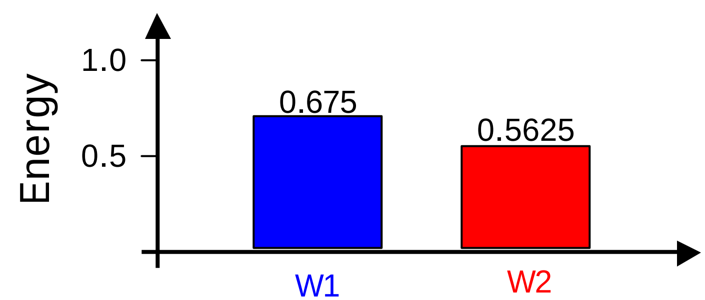
# Energy Performance Tradeoff

- Averaging is energy efficient

$$\frac{r_1^2+r_2^2}{2} \ge \left(\frac{r_1+r_2}{2}\right)^2 \rightarrow \overline{E(r)} \ge E(\bar{r})$$

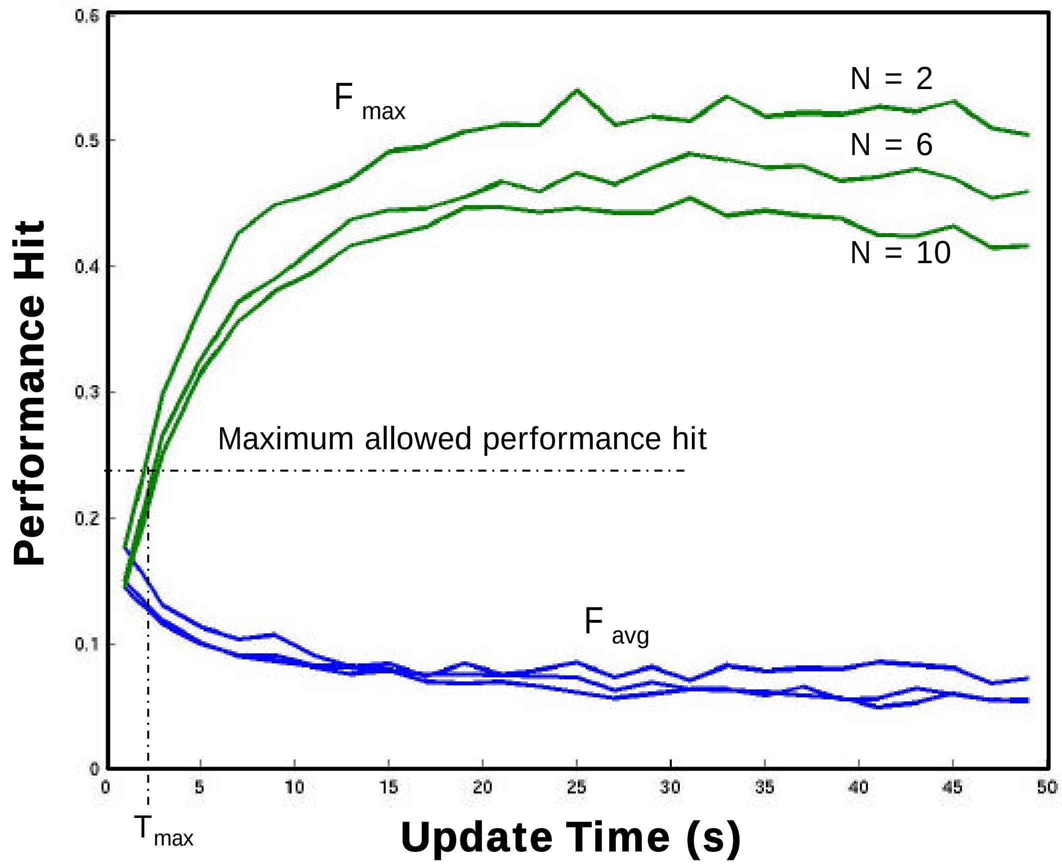



- Update time T depends on
  - Maximum allowed performance hit
  - DC/DC converter and frequency change overheads

# Performance Hit Metric




- Performance Hit Function

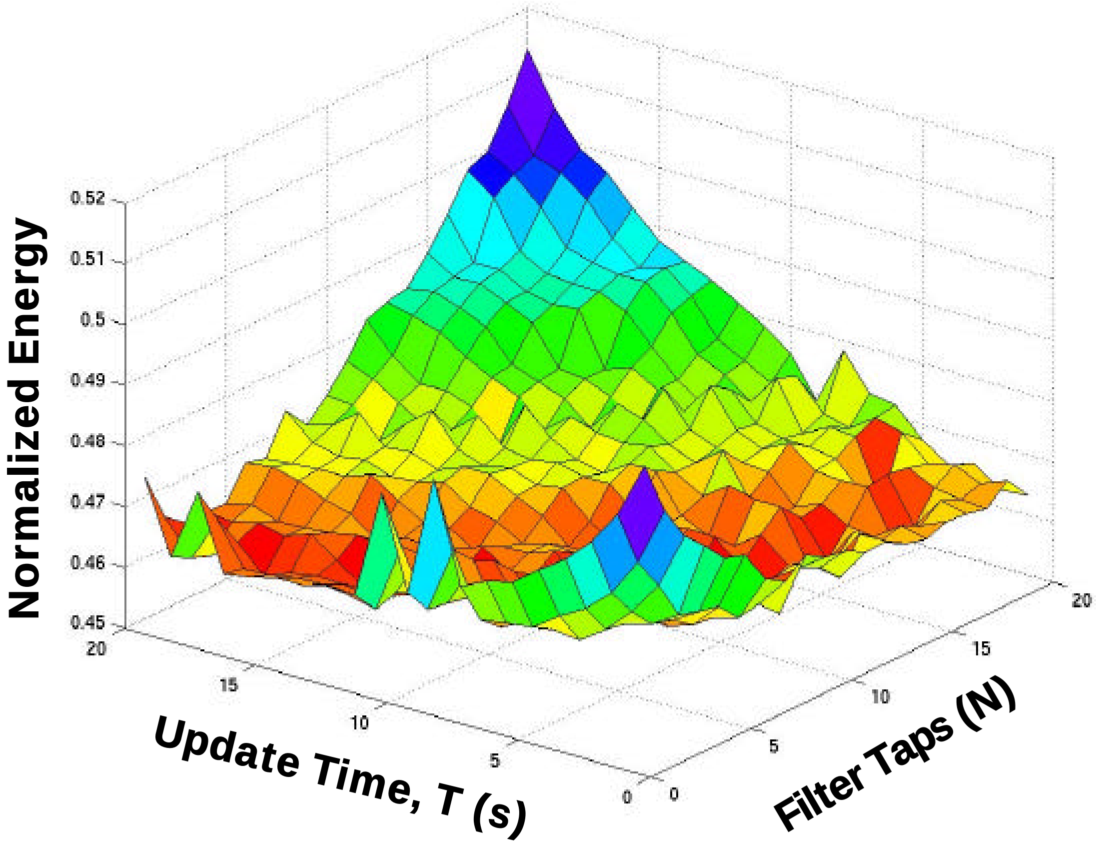
$$f(\Delta t) = \frac{\overline{w}_{\Delta t} - \overline{r}_{\Delta t}}{\overline{r}_{\Delta t}}$$

- Maximum and Average

$$f^{T}_{\max}(\Delta t), f^{T}_{avg}(\Delta t)$$

Maximum can be used set update time

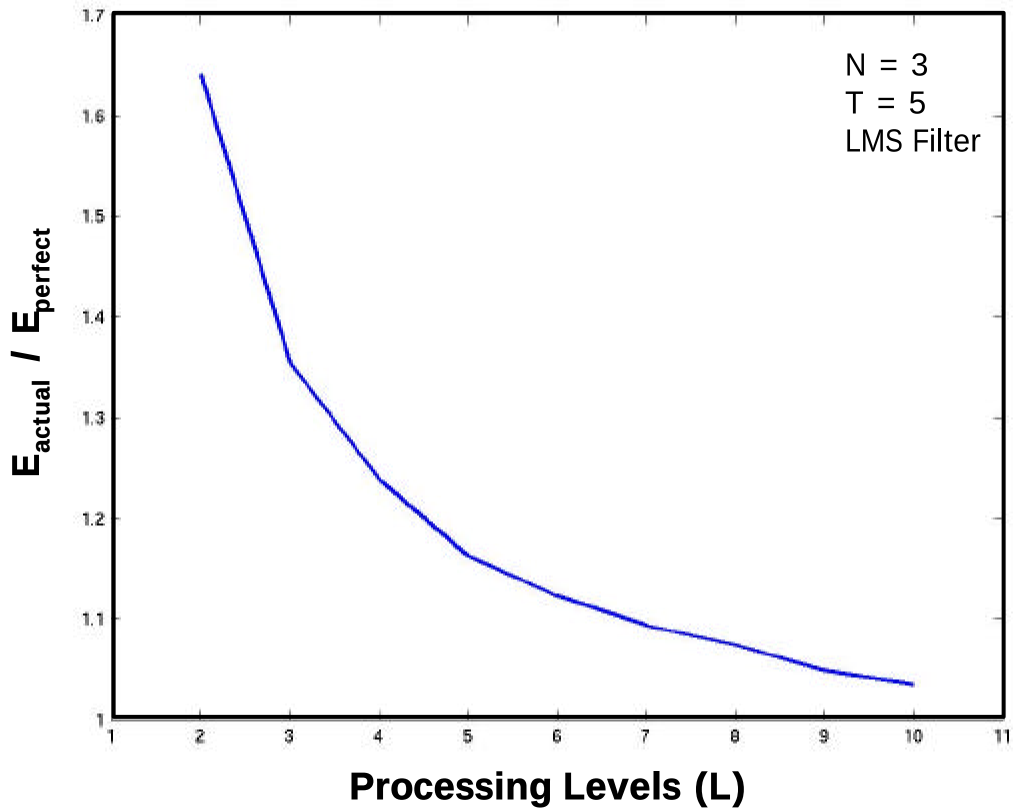
# Optimum Update Time and Taps




- Good choice
  N = 3
  T = 5 s

- N, T selections are not completely independent!

# Discrete Processing Levels




- Discrete frequency levels are not too bad.
- StrongARM has 11 levels [ degradation < 5% ]

# Results

| Trace | Filter | Energy Savings Ratio (ESR) | | | ESR Comparison | | $F_{avg}$ (%) | $F_{max}$ (%) |
|---|---|---|---|---|---|---|---|---|
| | | Max | Perfect | Actual | Max / Perfect | Perfect / Actual | | |
| Dialup Server | MAW | 2.9 | 2.4 | 2.2 | 1.2 | 1.10 | 10.6 | 34.8 |
| | EWS | | | 2.1 | | 1.11 | 10.8 | 36.3 |
| | EWA | | | 2.2 | | 1.09 | 10.6 | 35.4 |
| | LMS | | | 2.3 | | 1.03 | 14.7 | 43.1 |
| File Server | MAW | 76.7 | 23.5 | 16.7 | 3.3 | 1.41 | 12.6 | 42.8 |
| | EWS | | | 15.7 | | 1.50 | 7.4 | 33.8 |
| | EWA | | | 16.7 | | 1.41 | 9.2 | 37.4 |
| | LMS | | | 19.6 | | 1.20 | 14.1 | 47.7 |
| User Work-Station | MAW | 445.9 | 275.2 | 52.7 | 1.6 | 5.22 | 3.6 | 35.3 |
| | EWS | | | 59.5 | | 4.63 | 3.8 | 35.1 |
| | EWA | | | 52.1 | | 5.28 | 3.7 | 35.6 |
| | LMS | | | 53.0 | | 5.19 | 3.9 | 36.0 |

# Conclusions

- DVS is very effective for energy reduction
  - Upto 2 orders of magnitude savings possible
  - About 30% ‘instantaneous’ performance loss
- Averaged workloads are best
  - Makes system sluggish to workload changes
  - Unknown a priori
- Energy Performance Tradeoff
  - Faster updates lower visible performance loss
  - Faster updates also mean increased energy
- Workload prediction is crucial
- Adaptive LMS filtering is quite effective