

Our goal: Optimize Power Efficiency

Adapt voltage and frequency to the characteristics of applications

- Detect memory intensiveness **at runtime**
- Predict impact of DVFS
- Select optimal frequency

Model Performance/Energy under different frequencies

- Approximate models using Performance Counters

- Implement **Frequency Governors**

Reduce Energy without **harming** Performance by exploiting Slack

- Optimize EDP, ED²P, user-specific policies

Fine-Grained DVFS management

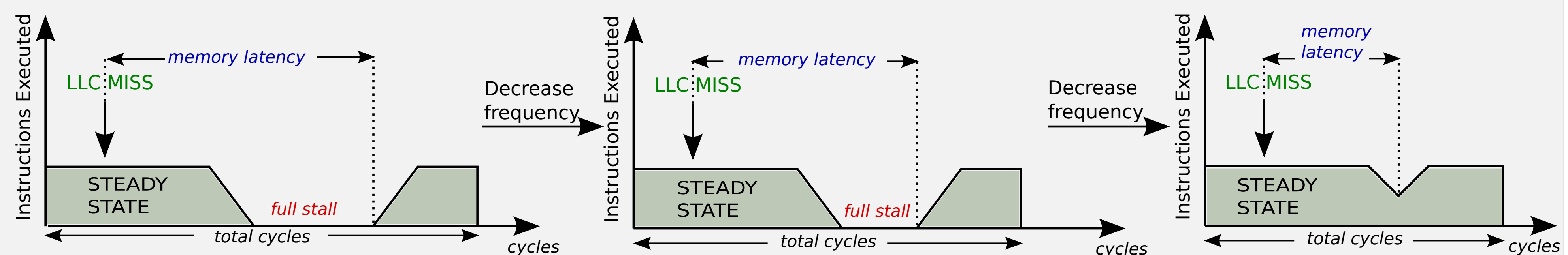
Exploit Memory Slack

Access time of main memory is not affected by processor Frequency Scaling

- Performance of memory-bound applications is less susceptible to Frequency Scaling
- Memory latency **measured in processor cycles** is reduced with processor Frequency Scaling

Reduce stalls (in cycles) by scaling down Frequency

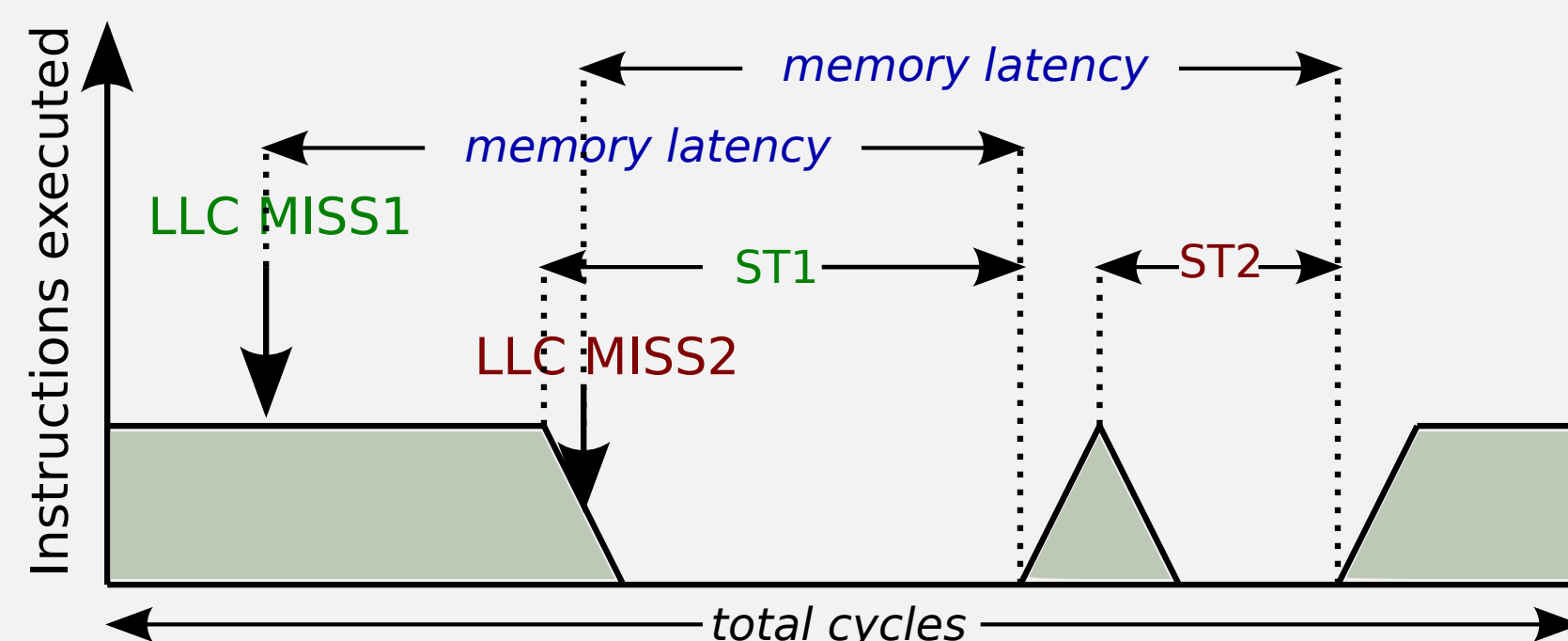
- But execution time of non-stall intervals is increased
- Need models to estimate the impact of Frequency Scaling in Performance



Analytical Interval-based DVFS Performance Models

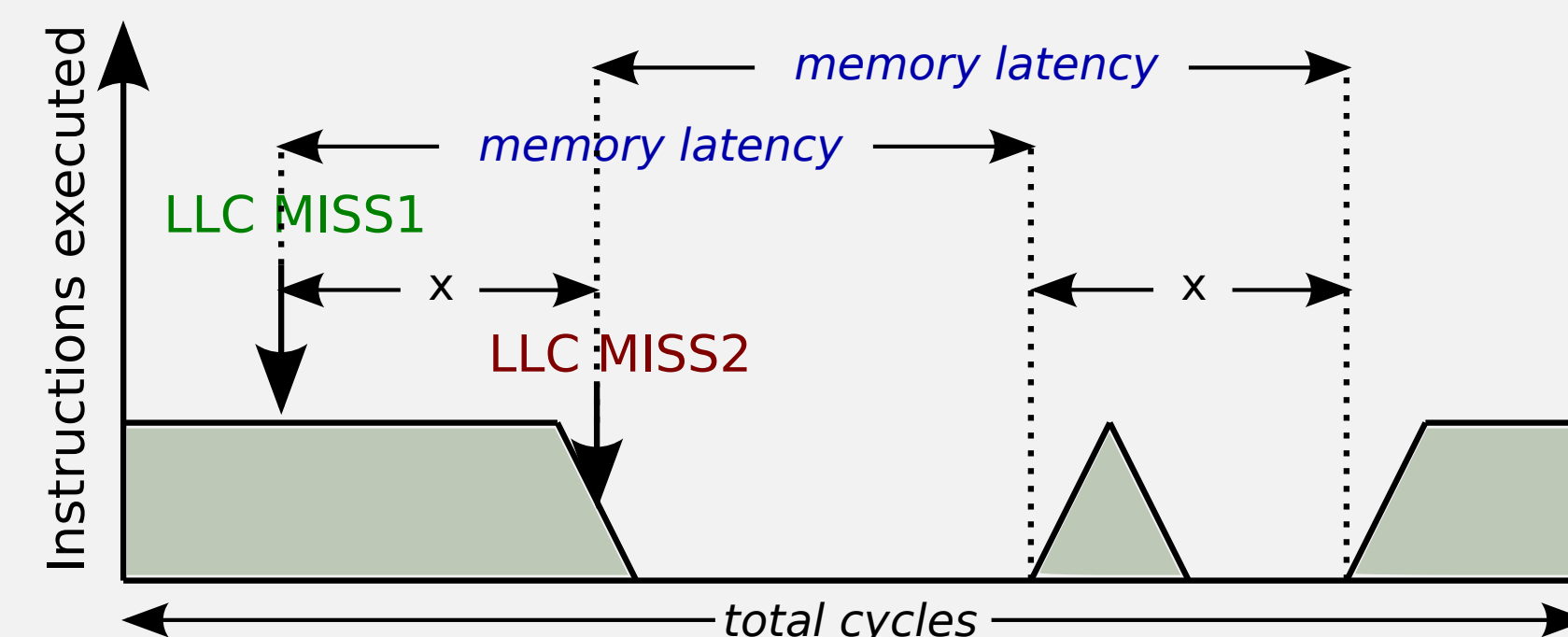
Stall-based Model

- Total stalls due to LLC misses (ST1+ST2) ~ frequency
- not completely true: **memory latency** ~ frequency and ST1+ST2 ≠ memory latency
- Accurate enough/easy to approximate with current Monitoring Hardware



Miss-based Model

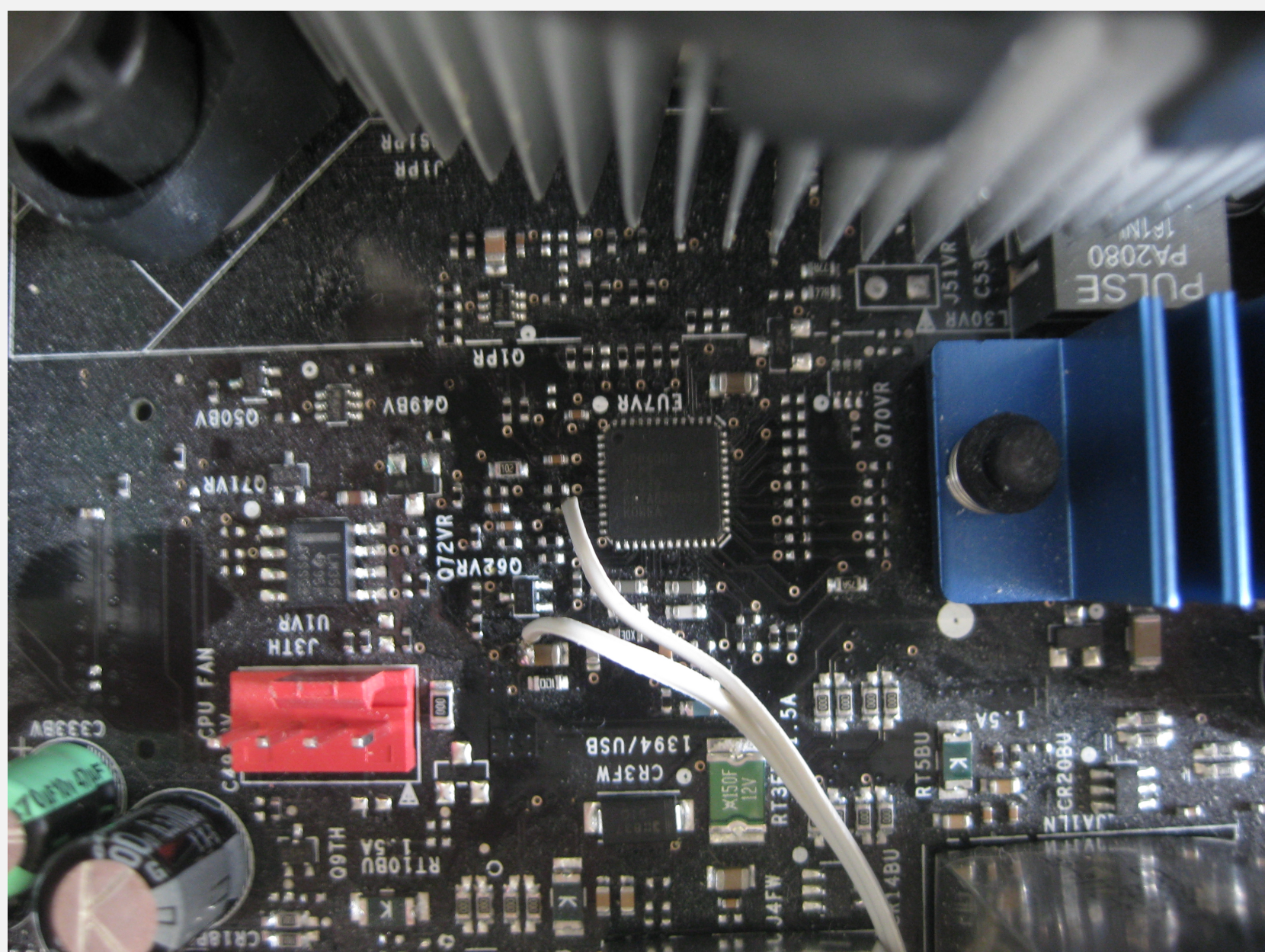
- If LLC MISS2 occurs x cycles after LLC MISS1 → serviced x cycles after LLC MISS1
- **only miss interval of the first miss in a cluster of misses scales with frequency**
- Very accurate/no **clusters of misses** event in current Monitoring Hardware



Measuring Power

Real hardware power measurements

- V, I measured directly from motherboard Voltage Regulator
- High resolution measurements



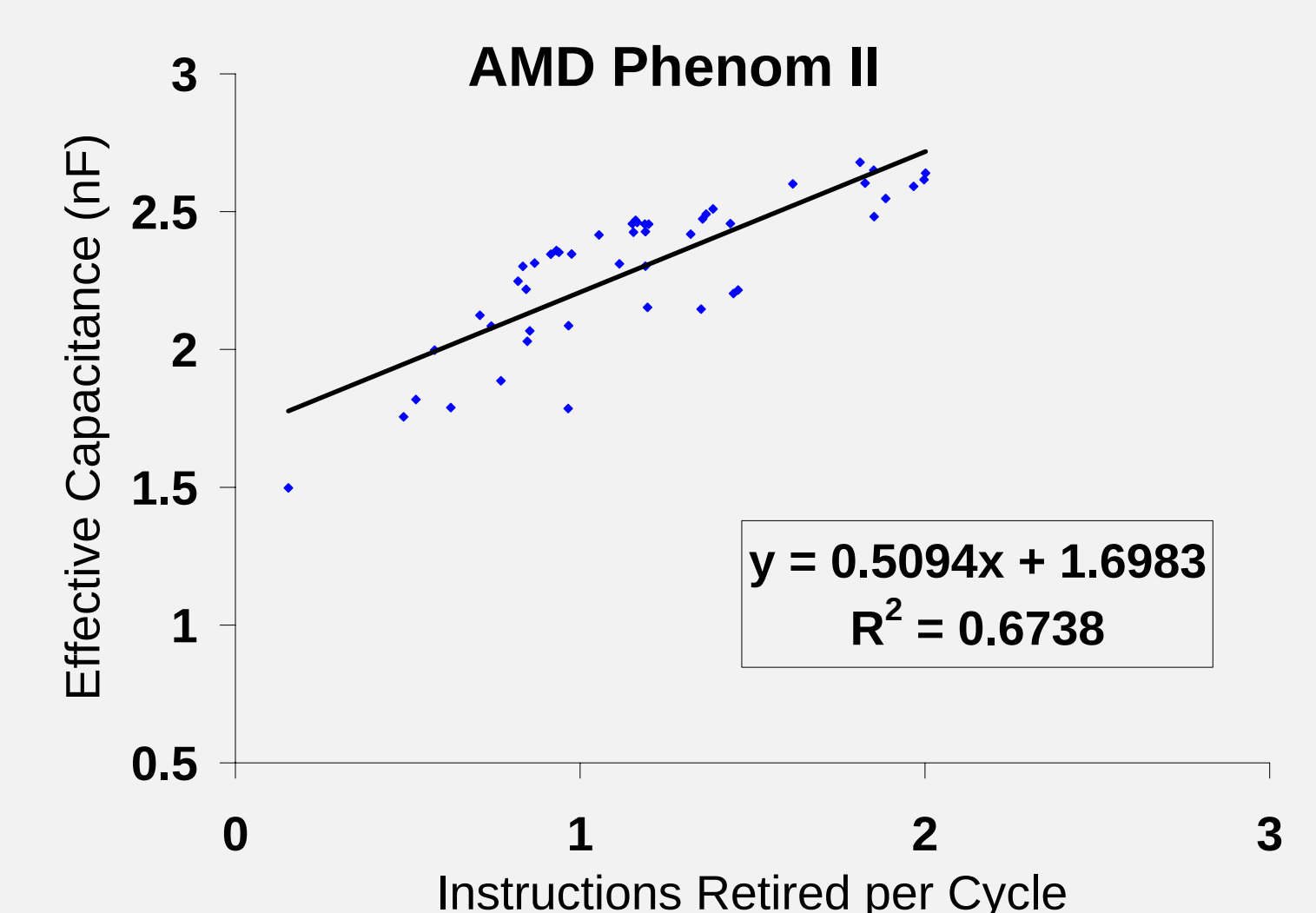
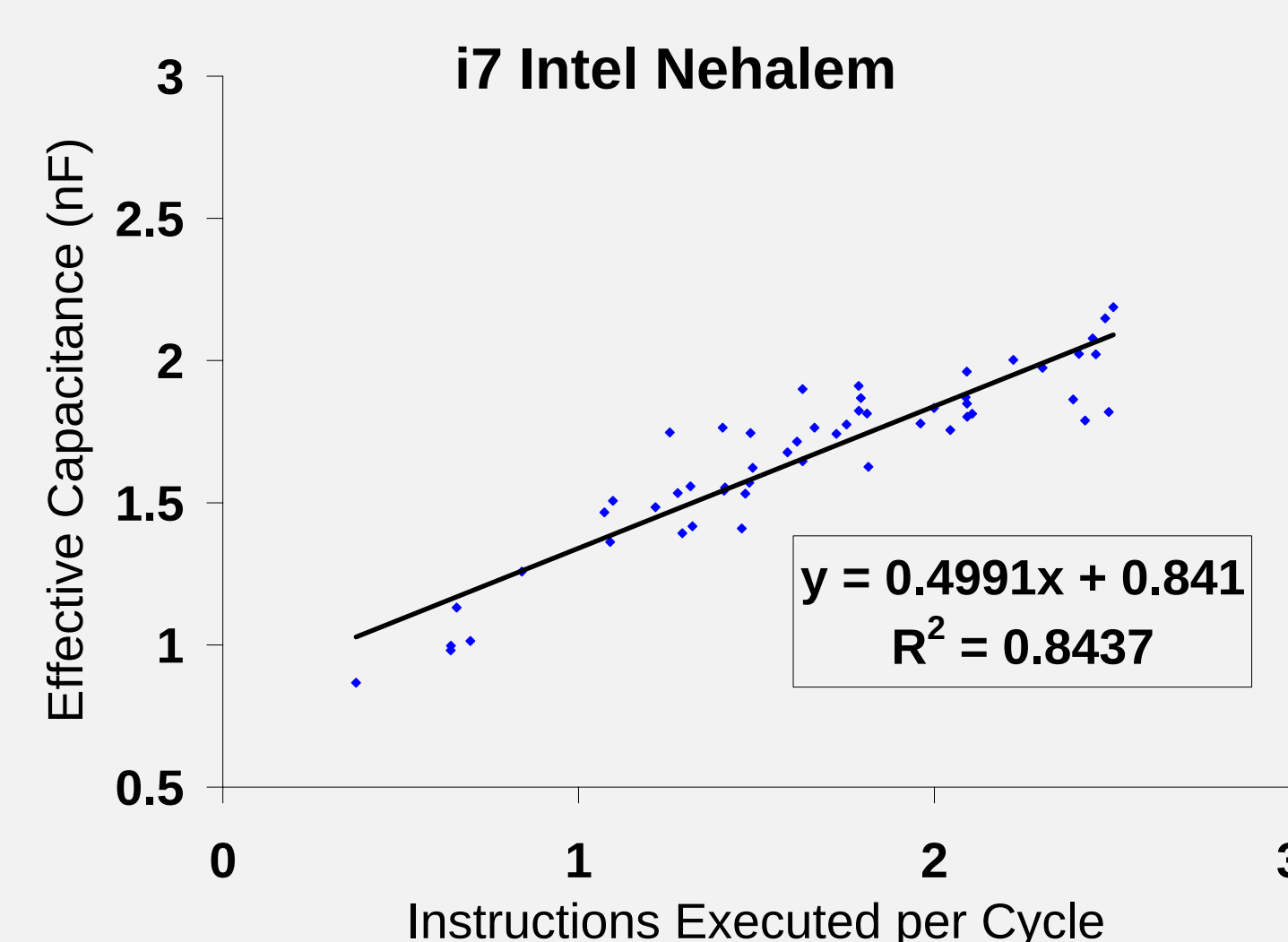
Modeling Power

$$\text{Dynamic Power} = \text{Frequency} \times \text{Effective Capacitance} \times \text{Voltage}^2$$

Frequency/Voltage known, but **Effective Capacitance depends on the application**

Correlate processor Effective Capacitance with

- Executed Micro-ops → Includes speculative execution (Intel Core i7)
- Retired Micro-ops → No speculative execution (AMD Phenom II)



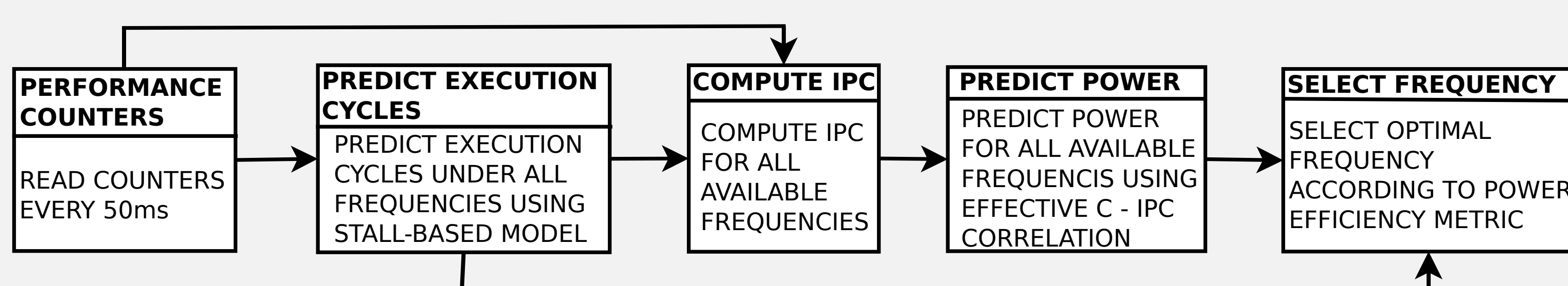
Linux Frequency Governors

Combine Performance/Power Models to implement Frequency Governors

- Fine-Grained management: limited by OS ticks (10ms)
- Software-only solution: estimate Performance and Power using Performance Monitoring Hardware
- Approximately optimal decisions (<2% error)
- Low overhead (<<1%)

Different Policies

- Optimize Power Efficiency metrics (EDP, ED²P)
- Policies by setting Performance Constraints
- Multicore management



Experimental Results

