A Framework for Dynamic Energy Efficiency and Temperature Management (DEETM)

Michael Huang, Jose Renau, Seung-Moon Yoo, Josep Torrellas
University of Illinois at Urbana-Champaign
http://iacoma.cs.uiuc.edu/flexram

Motivation

Alpha Processors History

*EV*4	*EV*5	*EV*6	*EV*7	*EV*8
30*W*	60*W*	90*W*	100*W*	150*W*

SIA Roadmap Hand Held

2000	2001	2002	2003	2004	2005
1.6W	1.7W	2.0W	2.1W	2.3W	2.4W

Motivation

- Energy Efficiency
  - Battery capacity improvement 10%/year
  - Electricity is expensive (10% USA energy budget)
  - 3B$ each year in idle electronics! (USA)
- High temperature
  - Meltdown, hotter than a hot-plate (0.6um)
  - EM: Electro Migration
  - DRAM Refresh, Leakage…

Current Common Approach

- Static optimizations
- Voltage Scaling
- Limitations of existing approaches
  - Coarse grain
  - Inefficient temperature control (sleep)
  - Independent targets: temp, energy efficiency

Goal

Unified framework for Dynamic Energy Efficiency and Temperature Management (DEETM)

- **Temperature**: enforce limit while minimizing slowdown
- **Energy efficiency**: maximize energy saving while exploiting performance slack

Contribution: DTEEM Framework

- Existing limitations:
  - static application → dynamic
  - coarse grain dynamic → fine grain
  - inefficient techniques → order techniques for maximum efficiency
  - independent targets: → unified target
    - temperature control
    - energy efficiency
- DEETM approach:
  - multiple techniques

DEETM Framework

- Monitors temperature & execution slack
- Runs decision algorithm periodically: {Thermal, Slack} components
- Activates low-power techniques
  - dynamically
  - incrementally
  - in prioritized order

Techniques

- Instruction filter cache
- Data cache subbanking
- Voltage scaling
- Voltage scaling: DRAM only
- Light sleep

Instruction Filter Cache

High power mode:

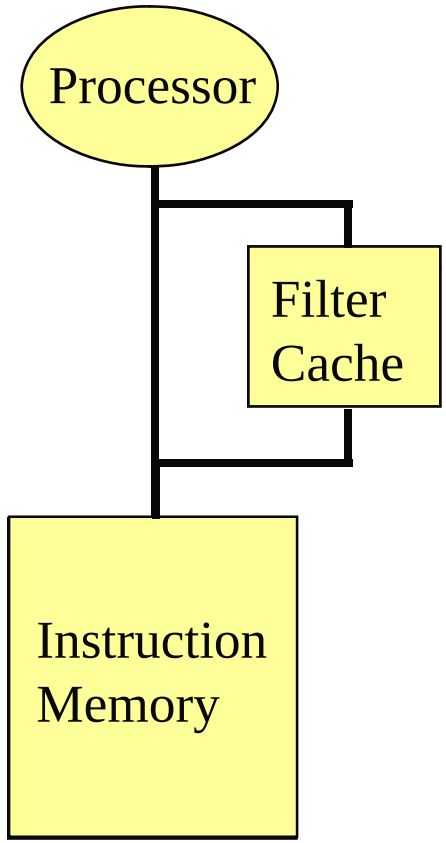
Time: 1

Energy: E

Low power mode:

Time: 1 + mr*1

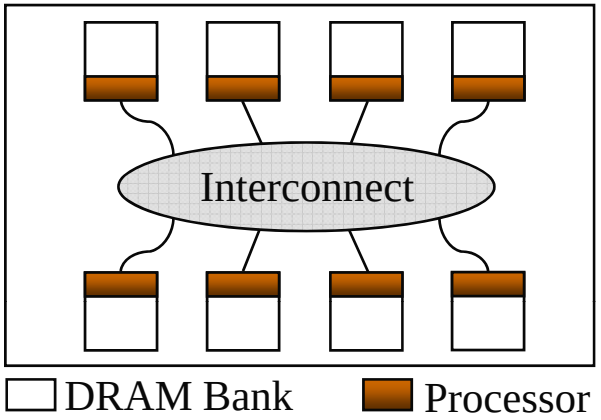
Energy: e + mr*E

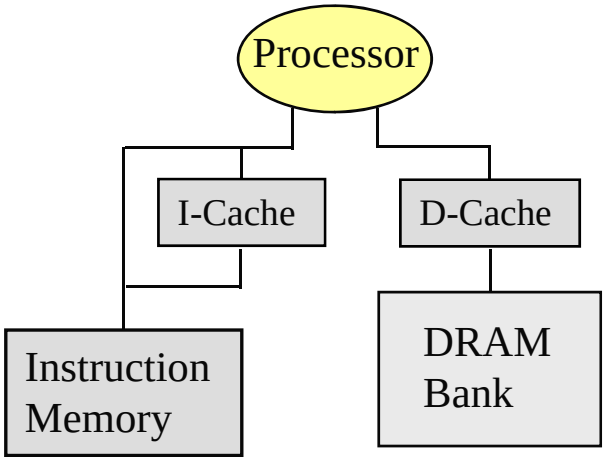


Techniques

- Instruction filter cache
- Data cache subbanking
- Voltage scaling
- Voltage scaling: DRAM only
- Light sleep

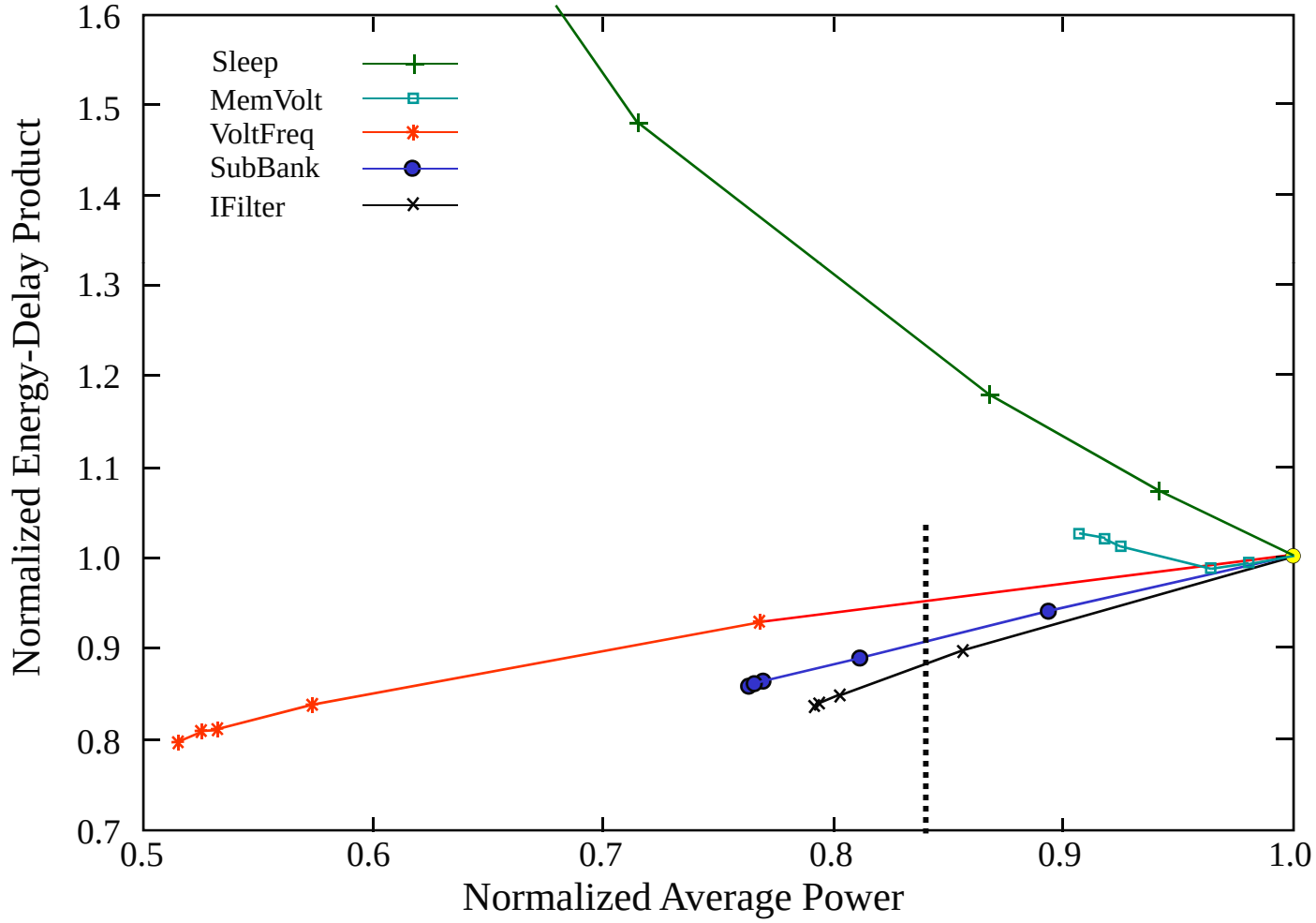
Chip Environment



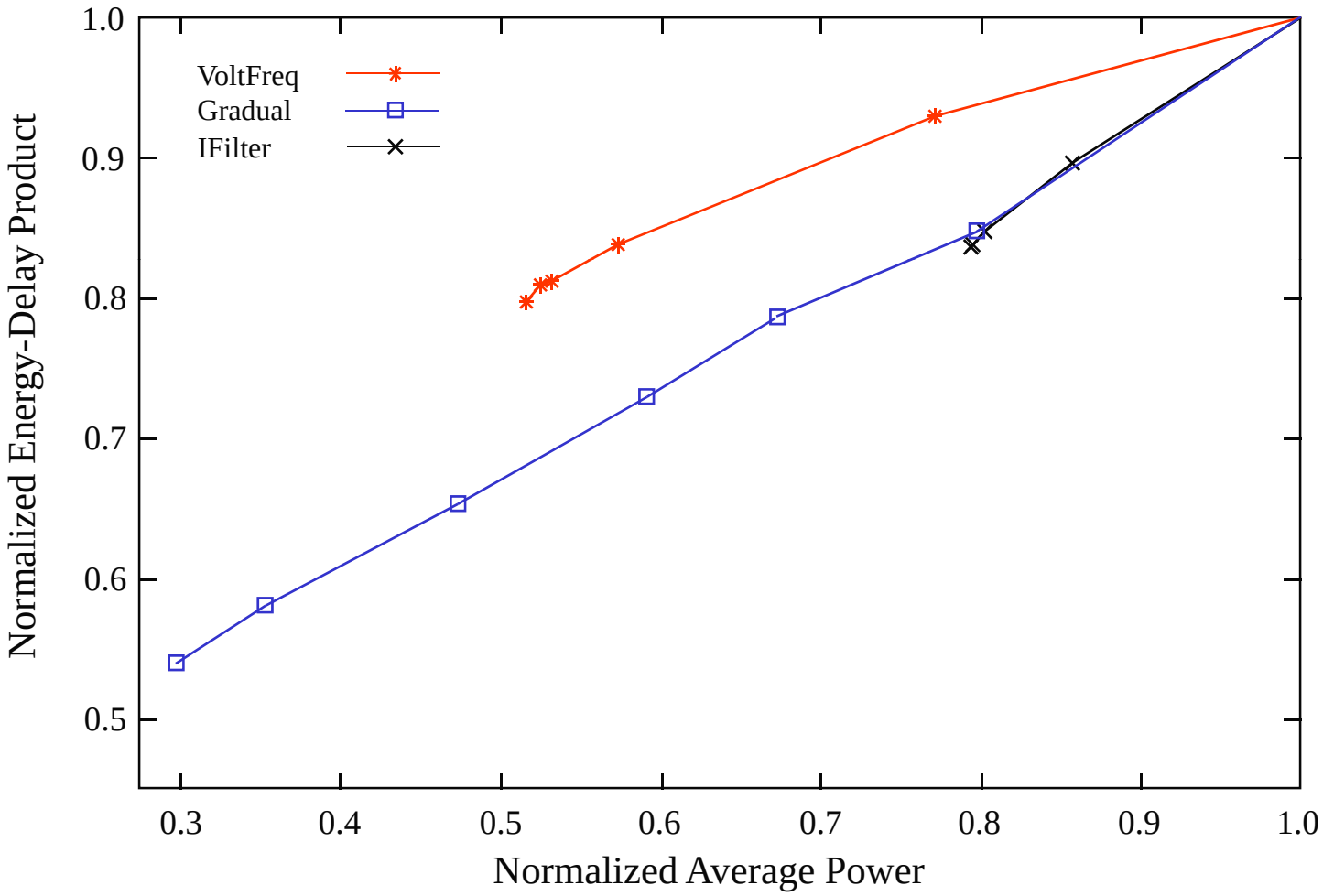


- Processor-in-memory
- 64 lean processors
  - 2-issue static
- Optimized memory hierarchy
- Integrated thermal sensors and instruction counter

Individual Techniques-E*D



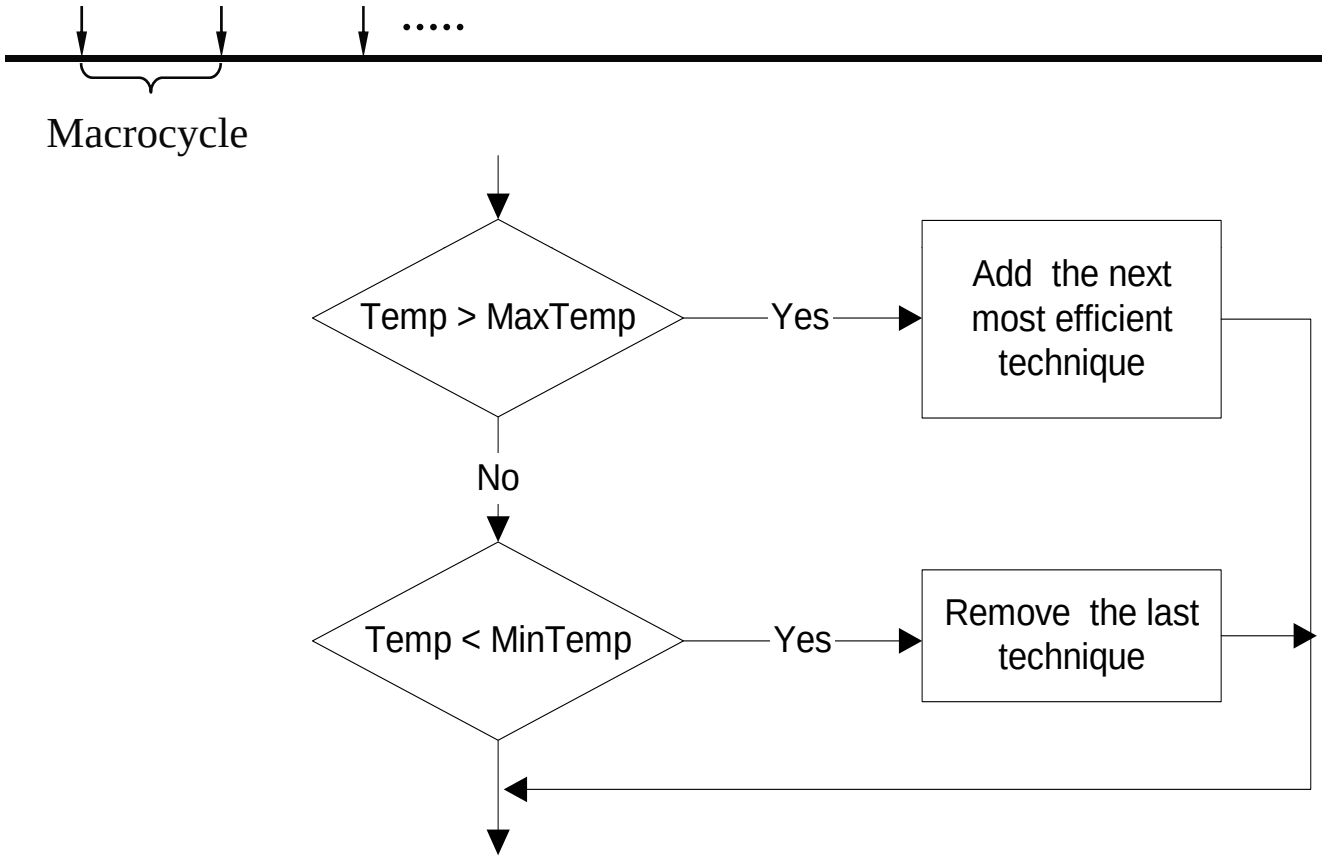
Combinations - E*D



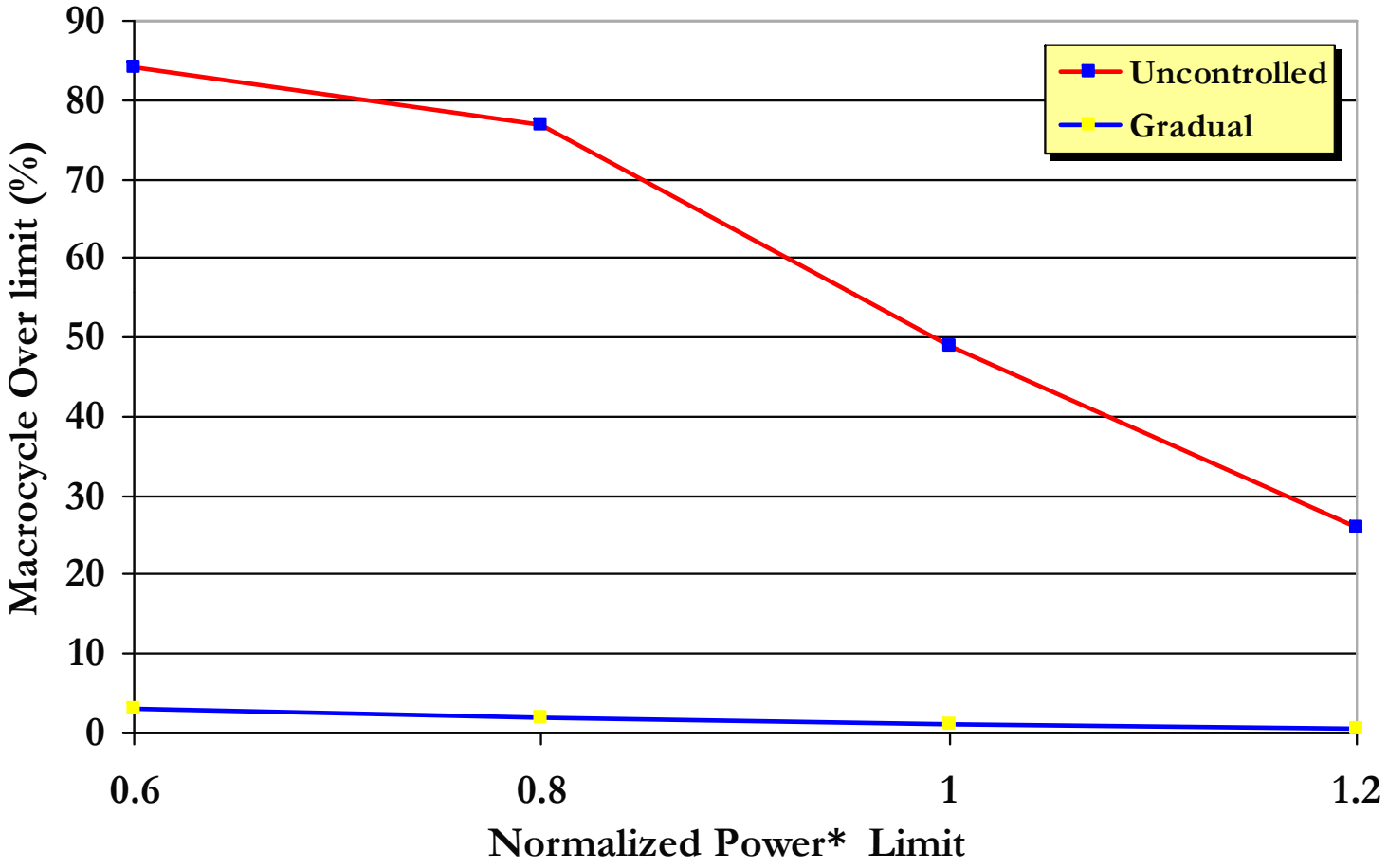
Classification Summary

- Techniques ordered by efficiency
  - *Thermal* and *Slack* have the same order
- System applies techniques in order
  - Dynamically and incrementally

Thermal Algorithm

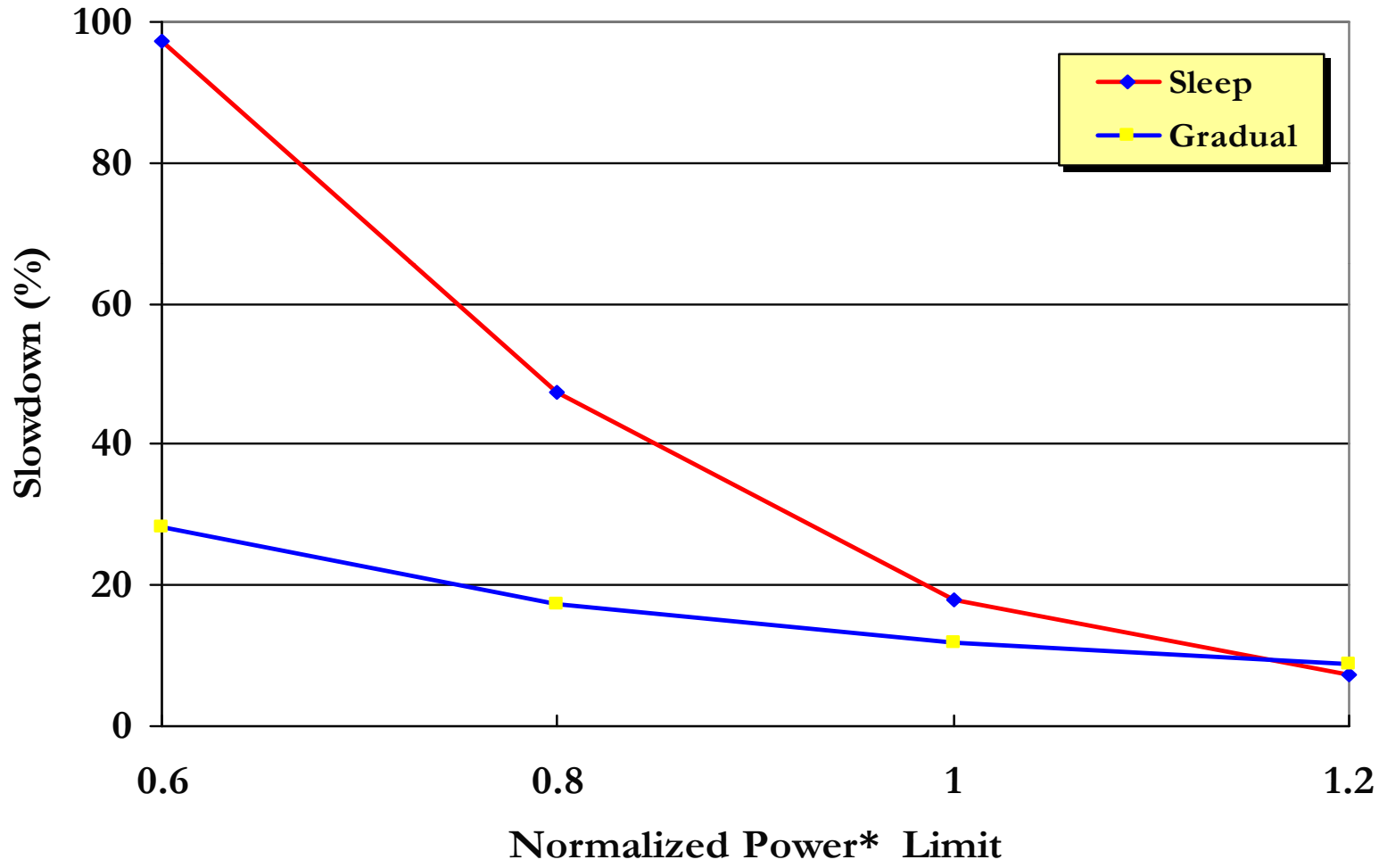


Temperature Control - Limit

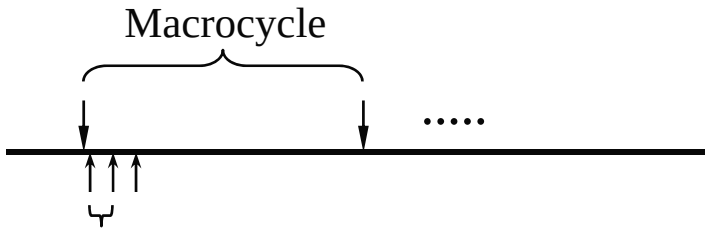


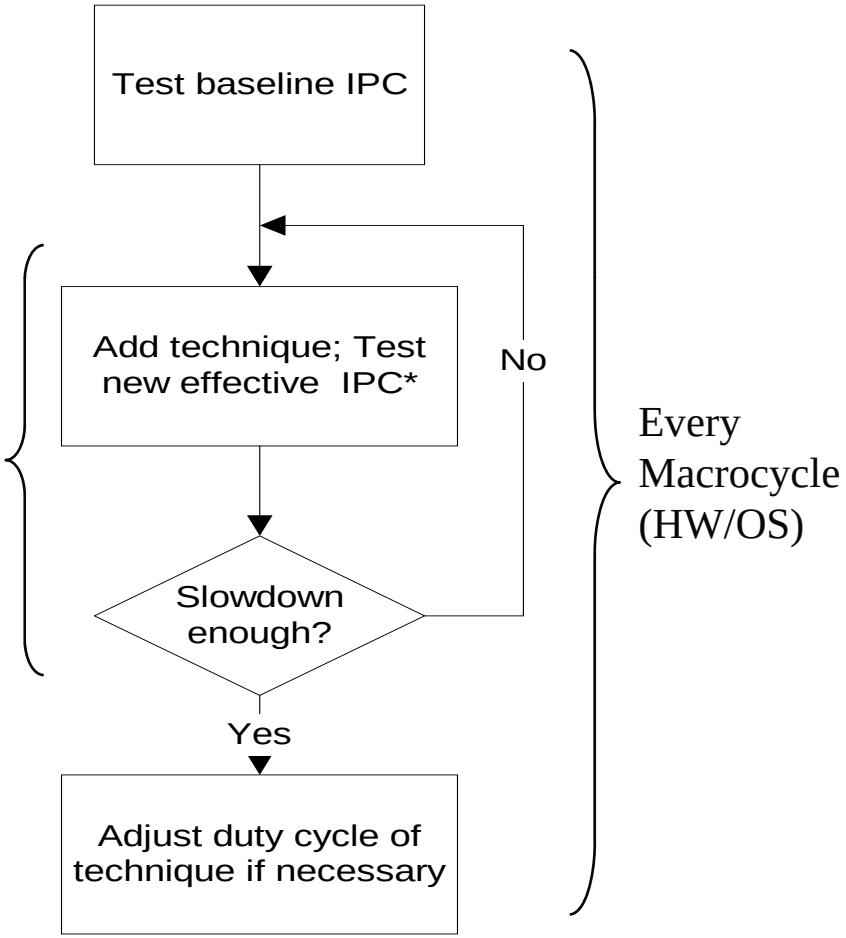
$$Power_i^* = 75\%Power_i + 25\%Power_{i-1}^*$$

Temperature Control - Slowdown



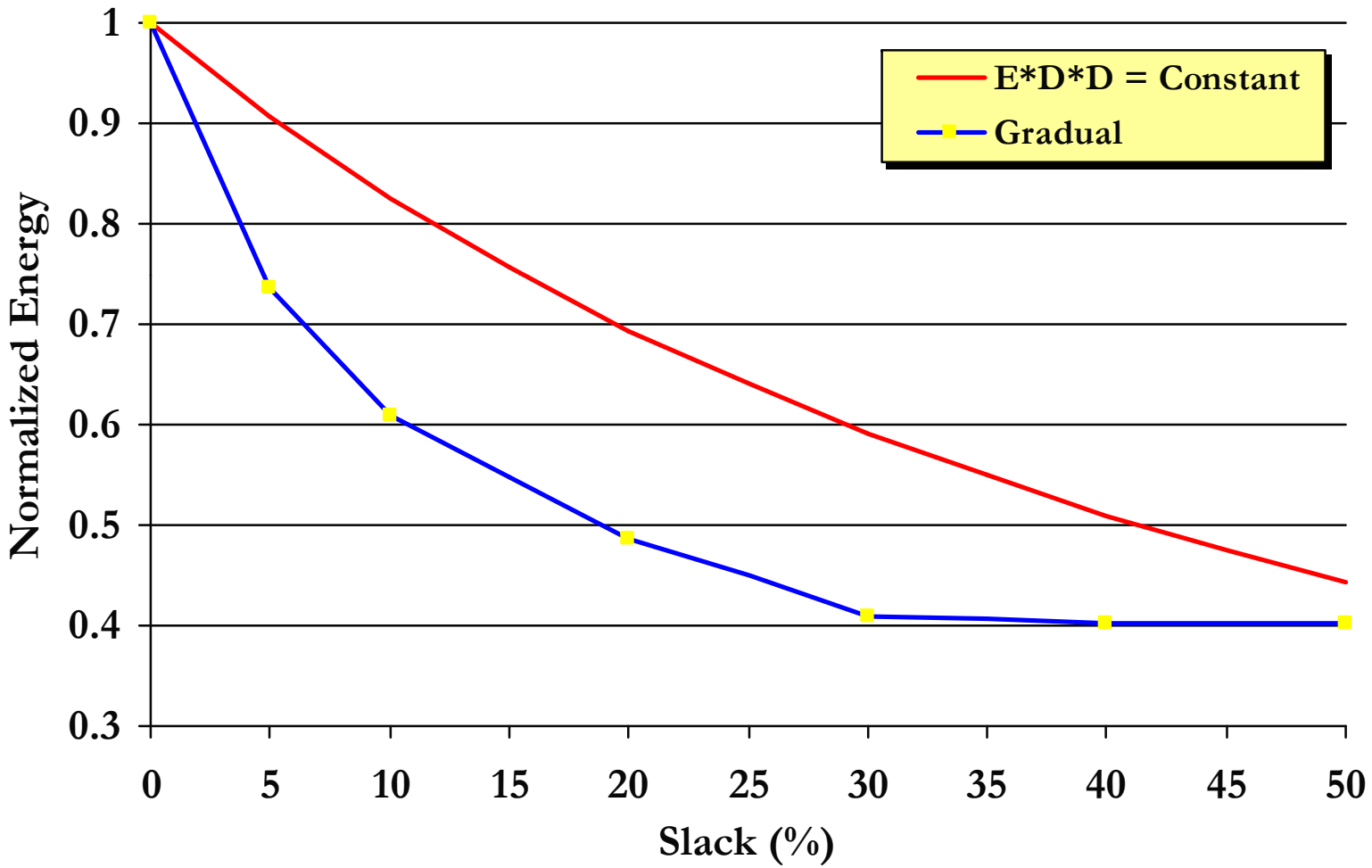
Slack Algorithm



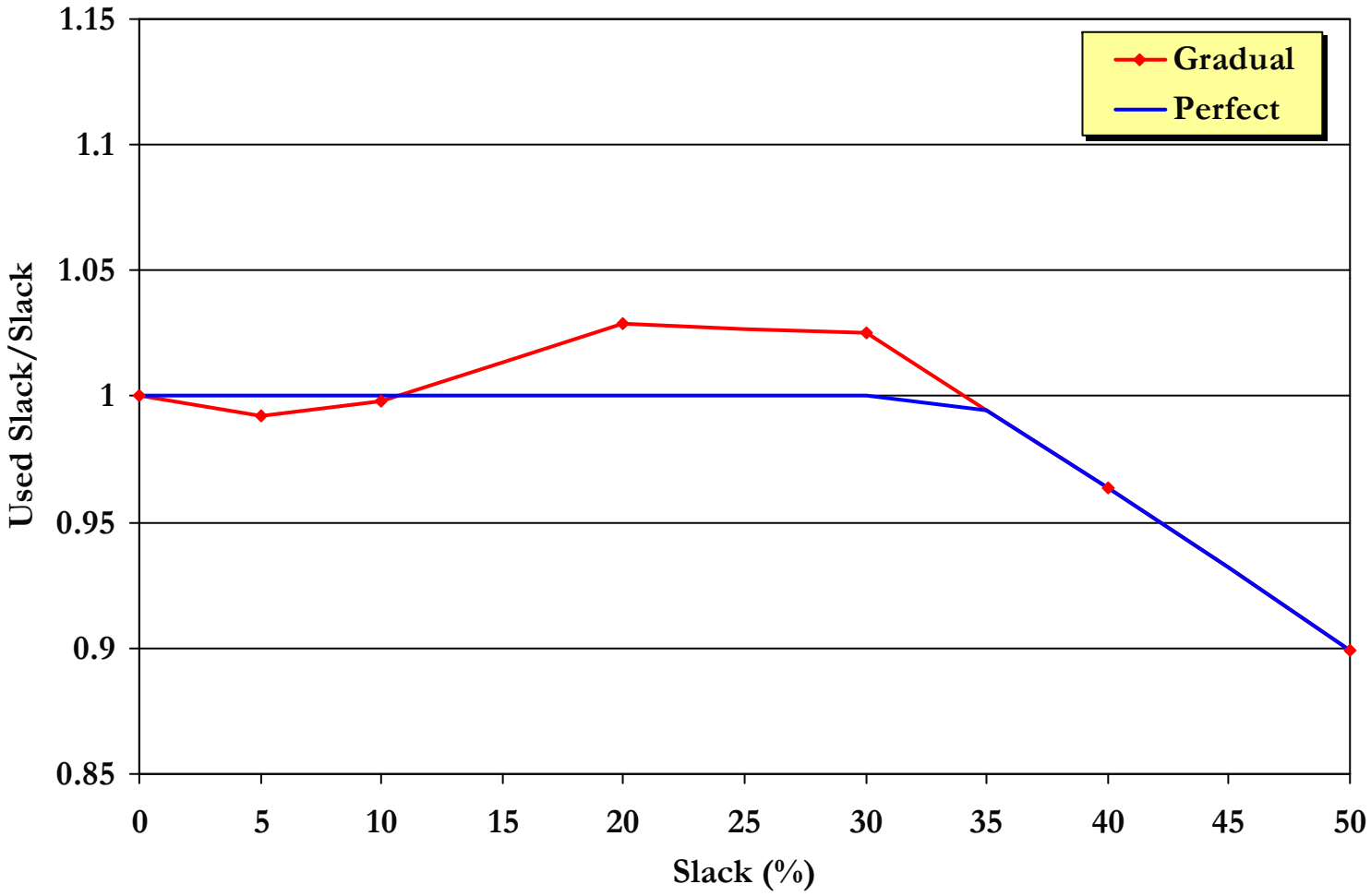


Effective IPC*: frequency-adjusted

Slack - Energy Consumption



Slack Misprediction



Also Covered in Paper

- Algorithm interaction
- Selecting macrocycles & microcycle
- Handling *Thermal Crisis* situation (thermal virus)
- Reducing technique activation overhead
- Flexible technique ordering
- Hardware vs. software implementation

Conclusions

- Effective & efficient temperature control
  - very few macrocycles still over limit
  - 27% longer execution vs. 98% (by sleeping)
- Efficient & accurate fine-grain exploitation of execution slack
  - 5% slack ⇒ 27% energy saved
  - small slack misprediction