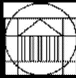
Presented at ISCA-30, June 2003
© 2003, ACM

Temperature-Aware Microarchitecture

Kevin Skadron, Mircea Stan

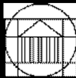
Wei Huang, Sivakumar Velusamy
Karthik Sankaranarayanan, David Tarjan



LAVA and HPLP Labs
University of Virginia

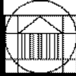
Talk Outline

- Why is "temperature-aware" design a topic of specific interest, and what does architecture have to do with it?
- Temperature-aware architecture requires an architectural model of temperature
  - Dynamic compact model: equivalent circuit based on thermal R and C
  - HotSpot is now publicly available on the web
- Architecture techniques for runtime thermal control (*dynamic thermal management*, DTM)
  - Localized response can outperform DVS
- Where do we go from here?

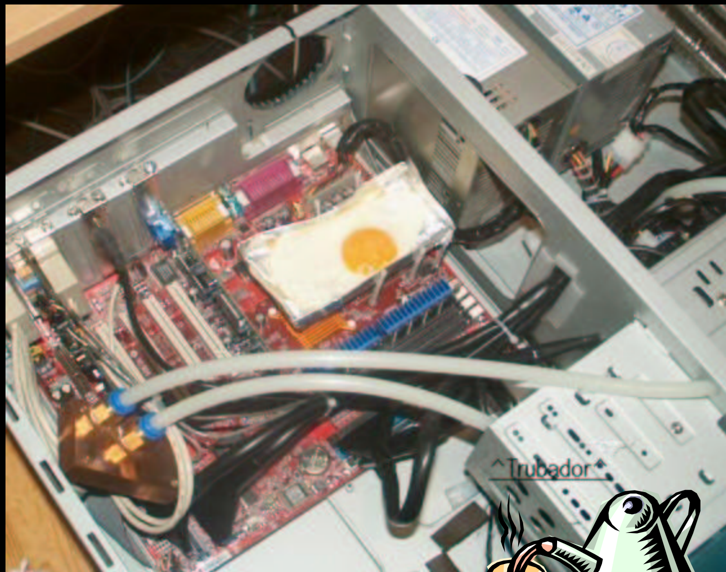


Metrics

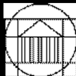
- Power consumption: first-order design constraint
  - average active power and idle power limit battery life, etc.
  - peak power limits power delivery (dI/dt), degrades battery
    - *not* well correlated with temperature
    - *not* the same as thermal design power
  - sustained power density limits thermal design/packaging
    - approx. same as thermal design power
  - Common fallacy: instantaneous, average power ≠ temperature
- Power-density is increasing exponentially
  - power density matters because this is proportional to the rate of heating per unit area
  - heating -- and therefore cooling costs -- are rising exponentially
  - currently $2-3 / W
- Need temperature-aware design!
  - Optimizing power is different than temperature



"Cooking-Aware" Computing?

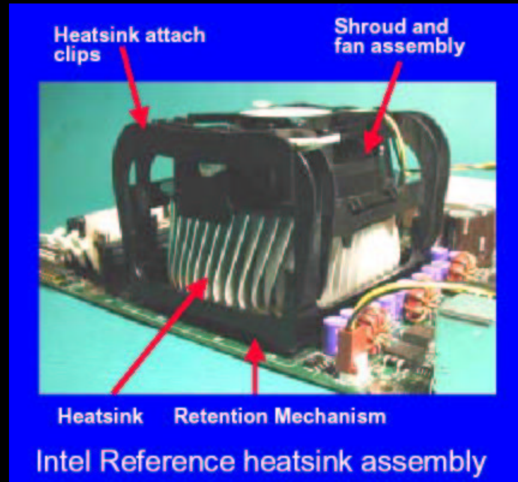


- Boiling water will come soon

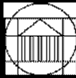


Thermal Packaging is Expensive

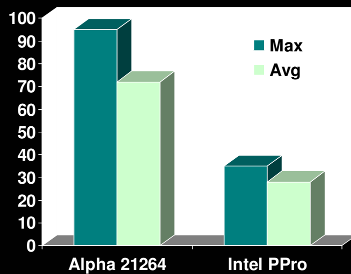
- P4 packaging



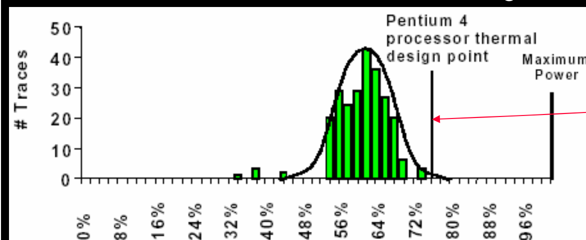
Source: Intel web site



Average vs. Worst Case Power → DTM



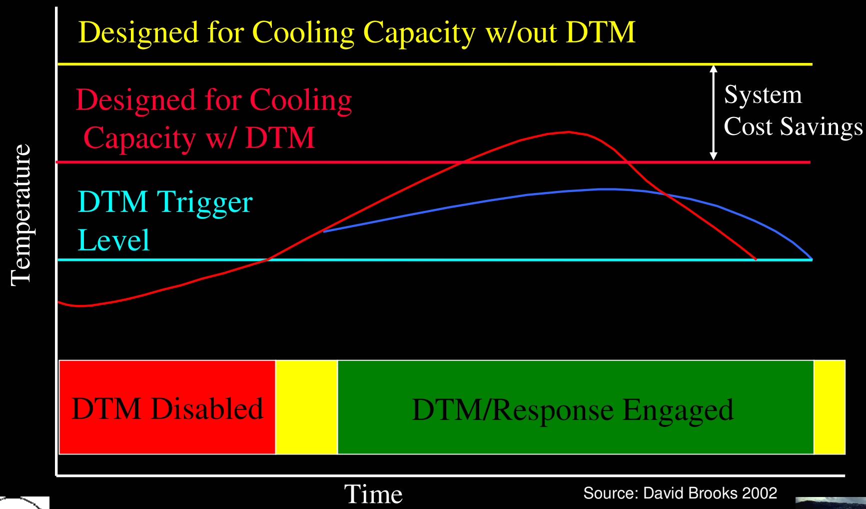
- *Average case* temperature can often be much lower than worst-case
  - Aggressive clock gating
  - Application variations
  - Underutilized resources, *e.g.,*
    - Low ILP
    - Floating point units during integer code execution
- Currently about a 30-40% difference
- Likely to further diverge…
- Target worst typical power dissipation



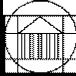
Gunther et al, ITJ 2001

Dynamic Thermal Management (DTM)

(Brooks and Martonosi, HPCA 2001)

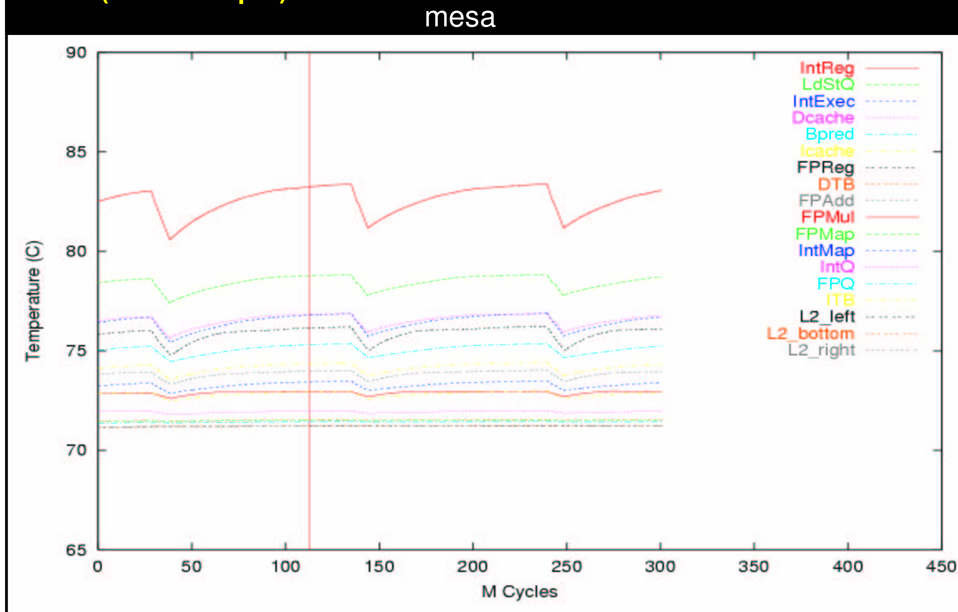


Source: David Brooks 2002



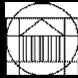
Time-Varying Behavior -- Example

(from HotSpot)



What Can Architects Do About This?

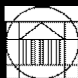
- Architectural blocks are a very natural granularity for thermal management
  - Use architectural responses to lower power density
  - Heating is localized--detect and respond to *hot spots*
  - Heating within a block is quasi-uniform
  - Heating is correlated with program behavior
  - Architects know how to manage this!
- The OS then provides knowledge of per-task thermal behavior and performance requirements
  - Task scheduling provides a great deal of flexibility
  - OS and architecture can cooperate
  - Architects know how to manage this too!



But how do we model heat in a practical way?

Talk Outline

- Why is "temperature-aware" design a topic of specific interest, and what does architecture have to do with it?
- Temperature-aware architecture requires an architectural model of temperature
  - Equivalent circuit based on thermal R and C
  - HotSpot is now publicly available on the web
- Architecture techniques for runtime thermal control
  - Localized response can outperform DVS
- Where do we go from here?



Thermal Modeling - Hot Spots

- Deal with “hot spots”
  - Localized heating occurs much faster than chip-wide
    - millisec. time scales
  - Chip-wide treatment is inaccurate
    - neglects hot spots
  - Power metrics are an unacceptable proxy
  - Temperature is sensitive to chip layout (floorplan)
  - Temperature is sensitive to details of thermal package

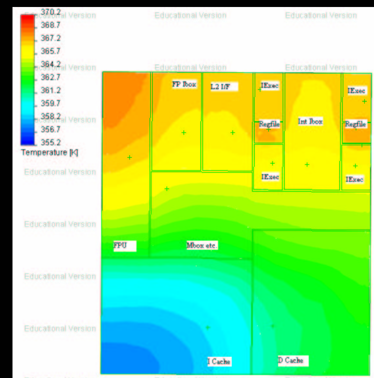
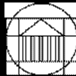
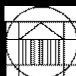


Image produced using Floworks



Thermal Modeling

- Want a fine-grained, dynamic model of *temperature*
  - A model that microarchitects and system architects can use
  - At a granularity that they can reason about
  - That accounts for adjacency and package
  - That is fast enough for practical use
- Averaging power dissipation is not accurate
  - Chip-wide average won’t capture hot spots
  - Localized average won’t capture lateral coupling
  - Does not account for block areas (*ie,* power density)
- ➔ HotSpot - a new model for localized temperature
  - Computationally efficient for use in power/performance simulators
  - Validated against FEM models (physical validation coming soon)
  - Publicly available

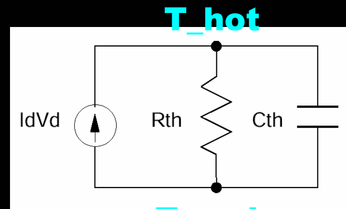


Simplistic Dynamic Compact Thermal Model

Electrical-thermal duality

- V ≅ temp (T)
- I ≅ power (P)
- R ≅ thermal resistance (Rth)
- C ≅ thermal capacitance (Cth)
- RC ≅ time constant

T_hot



T_amb

This is the model used by TEMPEST (Intel -- PACS'00, Dhodapkar *et al.)* -- it only models chip-wide temperature

KCL:

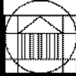
differential eq. $I = C \cdot dV/dt + V/R$

difference eq. $\Delta V = I/C \cdot \Delta t + V/RC \cdot \Delta t$

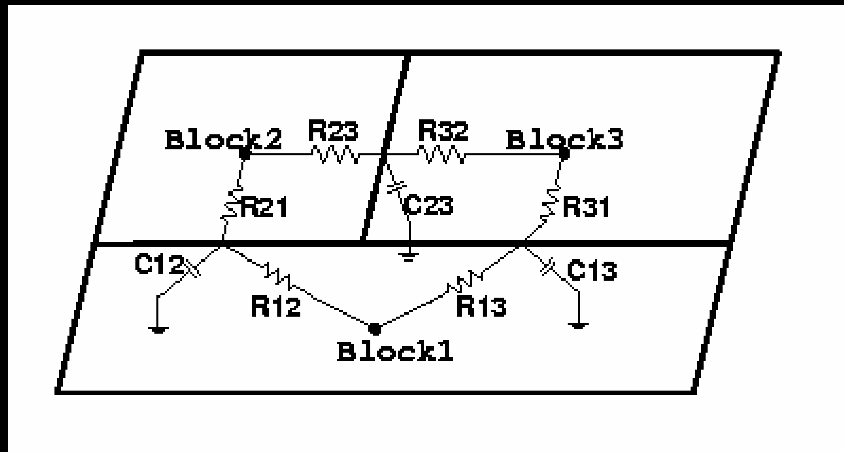
thermal domain $\Delta T = P/C \cdot \Delta t + T/RC \cdot \Delta t$

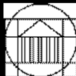
(T = T_hot – T_amb)

One can compute stepwise changes in temperature for any granularity at which one can get P, R, C

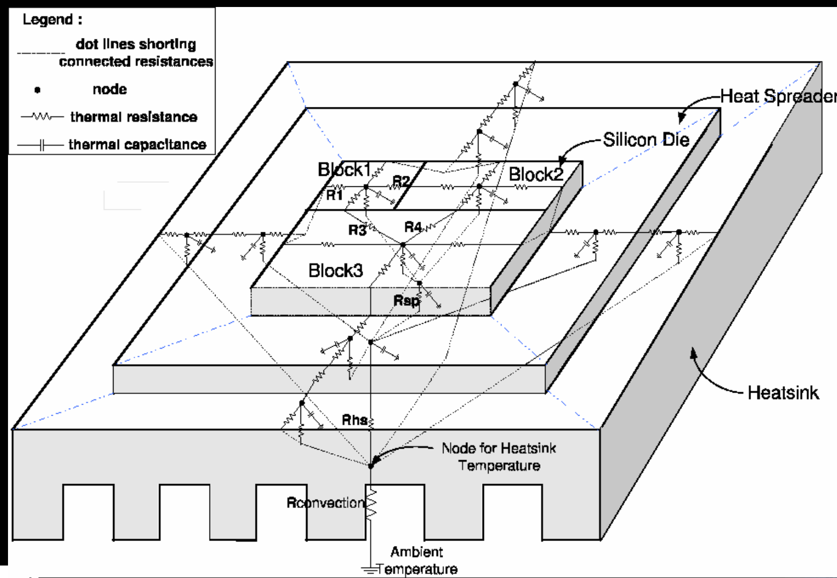


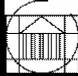
Our Model (Lateral)





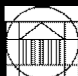
Our Model (Lateral and Vertical)

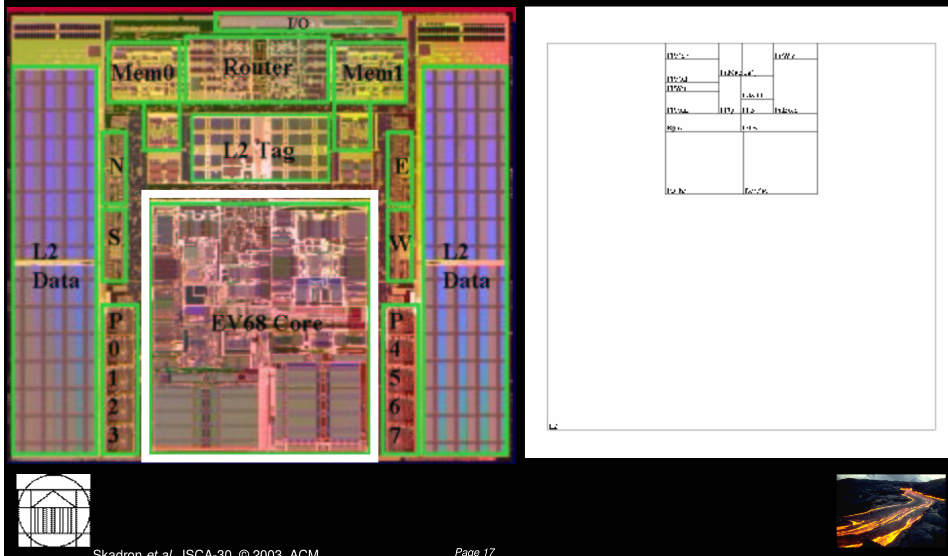


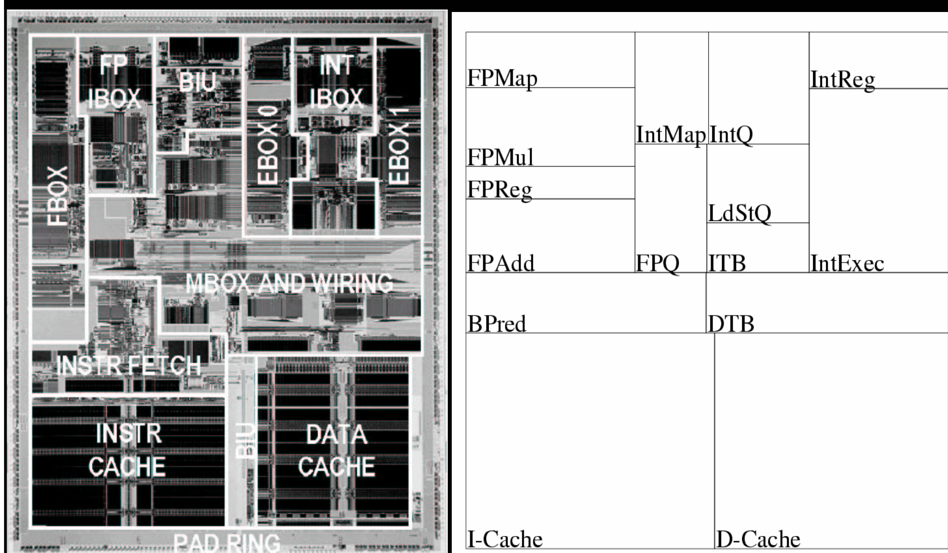


HotSpot

- Integrate HotSpot into a power/performance simulator
  - Time evolution of temperature is driven by unit activities and power dissipations averaged over 10K cycles
  - Power dissipations can come from any power simulator, act as “current sources” in RC circuit
  - Simulation overhead in Wattch: < 1%
- Requires models of
  - Floorplan: *important for adjacency*
    - Understanding of granularity vs. accuracy
  - Package: *important for spreading and time constants*
    - Combination of modeling, published numbers, and “custom” numbers (to obtain interesting behavior)
- Eventually need a model for the entire system “box”

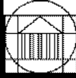



Floorplan We Use: Alpha 21364
I/O
Mem0
Router
Mem1
L2 Tag
N
E
S
W
L2
Data
EV68 Core
P
0
1
2
3
4
5
6
7
Skadron et al. ISCA-30 © 2003 ACM
Page 17


Floorplan – EV7 core
FP
IBOX
BIU
INT
IBOX
EBOX 0
EBOX 1
FBOX
MBOX AND WIRING
INSTR FETCH
INSTR
CACHE
BIU
DATA
CACHE
PAD RING
FPMap
FPMul
FPReg
FPAdd
IntMap
IntQ
LdStQ
FPQ
ITB
IntReg
IntExec
BPred
DTB
I-Cache
D-Cache

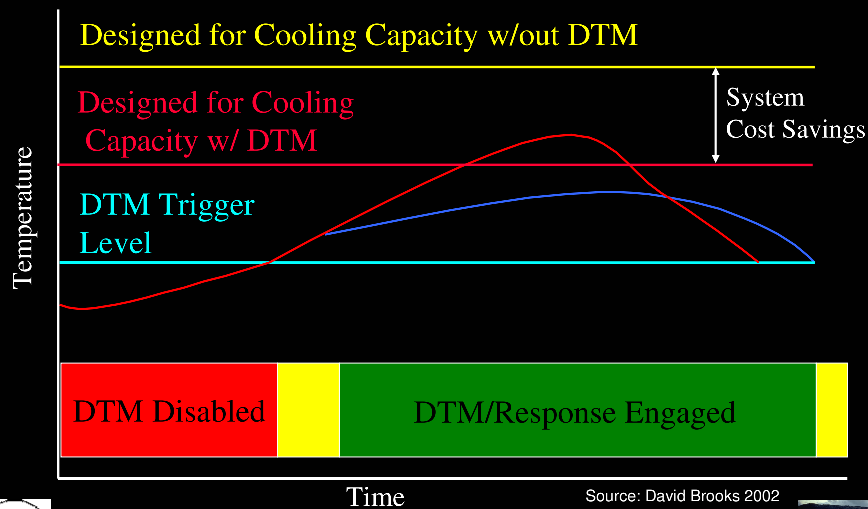
Talk Outline

- Why is "temperature-aware" design a topic of specific interest, and what does architecture have to do with it?
- Temperature-aware architecture requires an architectural model of temperature
  - Equivalent circuit based on thermal R and C
  - HotSpot is now publicly available on the web
- Architecture techniques for runtime thermal control
  - Localized response can outperform DVS
- Where do we go from here?

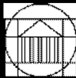


Skadron *et al.* ISCA-30 © 2003 ACM   *Page 19*

Dynamic Thermal Management (DTM)



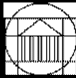
Source: David Brooks 2002



Skadron *et al.* ISCA-30 © 2003 ACM   *Page 20*

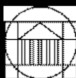
Previously Published DTM Techniques

- DVS
- Clock gating (Pentium 4, Gunther *et al.* 2001)
- Fetch gating or "toggling" (Brooks and Martonosi 2001)
  - Feedback-controlled fetch gating (Skadron *et al.* 2002)
- Fetch/decode throttling (Motorola G3, Sanchez 1997)
- Speculation control (Manne *et al.* 1998)
- Dual pipeline (Lim *et al. 2002)*
- Low-power caches (Huang *et al.* 2000)
  - DEETM framework used a hierarchy of response



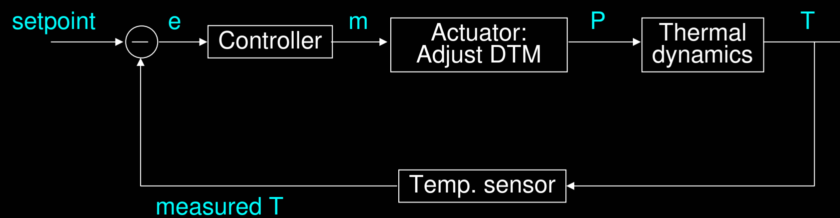
DTM Techniques – Comparison

- DVS (feedback-controlled) PI-DVS
  - Consider both 10 µs stall and 10 µs delay with no stall PI-DVS-i
- Clock gating (feedback-controlled) PI-GCG
- Local toggling (feedback-controlled) PI-LTOG
  - Domains: fetch, integer, FP, and ld-st
- Migrating computation MC
  - Spare integer register file
    - one extra cycle for register-file access
  - PI-LTOG as fallback
- "Temperature-Tracking" frequency scaling TT-DFS
  - Scale frequency linearly with temperature
    - 18% variation over the range 0-100° (Garrett & Stan)
    - No voltage scaling



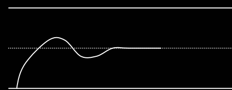
Feedback Control of DTM

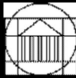
- Formal feedback control



PID: $u[k] = u[k-1] + K_c \cdot e[k-1]$

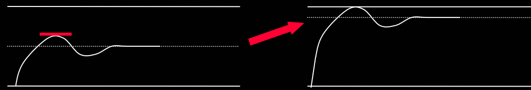
- easy to compute
- toggling = f(m)



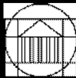


Benefits of Formal Feedback Control

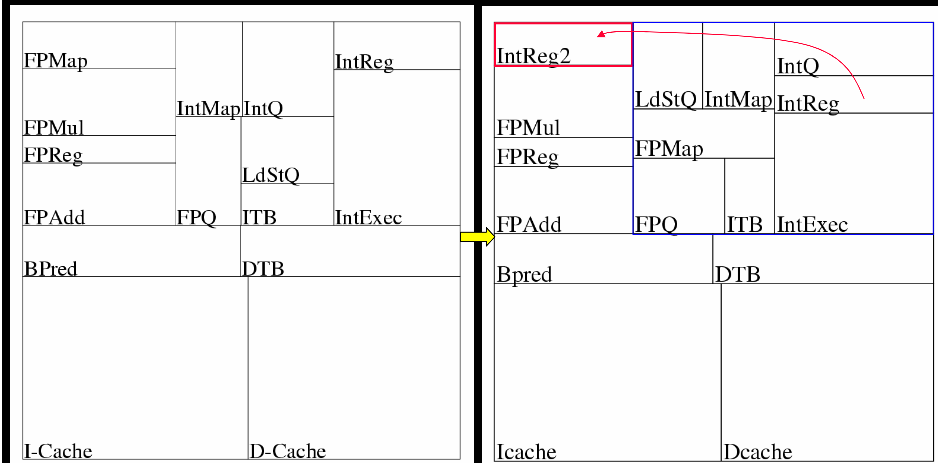
- Ideal for *regulatory* problems and autonomic control
- Regulatory: hold value to a specified setpoint
  - Example: temperature

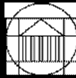


  - Proved that PI controller will not allow temperature to exceed setpoint by more than 0.2°
    - Max power dissipation, thermal dynamics, sampling rate ⇒ max overshoot
- Decades of theory to guide this; staple of EE



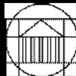
Migrating Computation



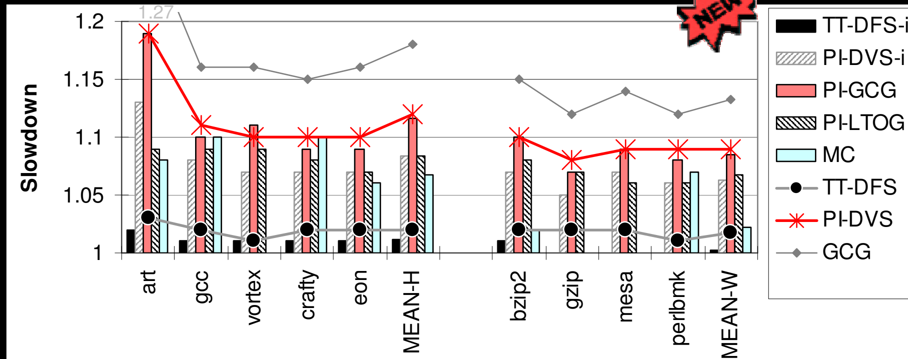


Simulation Details

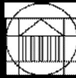
- 9 SPEC2000 benchmarks, both integer and FP
  - 4 hover near 81.8℃, rest are above
- SimpleScalar/Wattch, modified to model pipeline and power of an Alpha 21364 as closely as possible
- Scaled to 130nm, 1.3V, 3.0 GHz
- Die thickness: 0.5mm
- Ambient temperature (inside computer case): 45℃
- Rconvection = 0.8 K/W
  - 0.7 K/W necessary if DTM not available
- Max allowed temperature: 85℃
- Sensor imprecision: 2℃ fixed offset, 1℃ noise
- Trigger temperature/setpoint: 81.8℃



DTM Performance Comparison

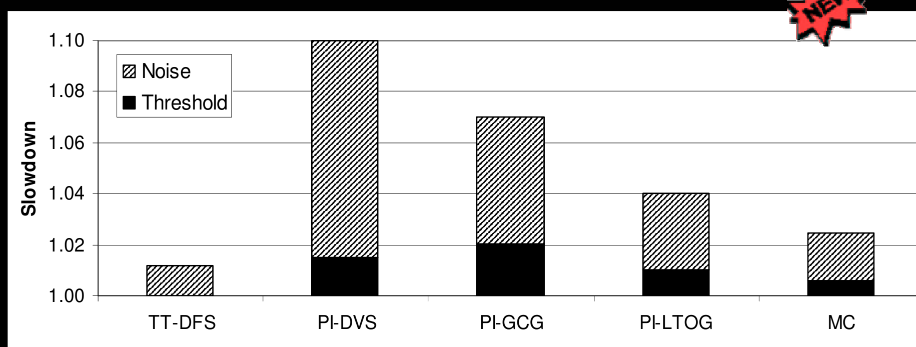


- TT-DFS is by far superior when timing is the only concern
- MC is slightly better than LTOG
- All three schemes outperform non-ideal DVS
- A hierarchy makes sense: *eg,* TT-DFS → MC → GCG
- For arch. simulation, initial heat-sink temp. matters
- Results very sensitive to worst-case assumptions

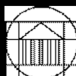


Skadron *et al.* ISCA-30 © 2003 ACM Page 27

Impact of Sensor Noise



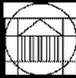
- Spurious DTM responses are the most costly
- This especially hurts non-ideal DVS due to stall associated with changing the setting



Skadron *et al.* ISCA-30 © 2003 ACM Page 28

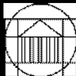
Talk Outline

- Why is “temperature-aware” design a topic of specific interest, and what does architecture have to do with it?
- Temperature-aware architecture requires an architectural model of temperature
  - Equivalent circuit based on thermal R and C
  - HotSpot is now publicly available on the web
- Architecture techniques for runtime thermal control
  - Localized response can outperform DVS
- Where do we go from here?



A Few Important Research Problems

- Understand tradeoff between performance cost and packaging cost
- Figures of merit, esp. independent of DTM
- Characterize thermal properties of different programs
  - Different application classes
  - Thermal interaction of different tasks (initial temperature effects)
- Understand how to balance DTM against real-time and other workload requirements
- Circuit - architecture - OS interactions & cooperation
- Understand thermal – energy tradeoffs

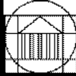


System-Level Issues

- Temperature-aware circuits (modify threshold, etc.) under micro-architectural control

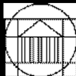
← microarchitecture →

- Temperature-aware OS
  - Use current operating temperature to guide scheduling
  - Mix time slices between hot and cold applications (*eg,* Rohou and Smith 1999)
  - Schedule to balance thermal stress across chip, system
  - Satisfy real-time or quality-of-service requirements
  - Match DTM to task priorities
- Use system-level knowledge to guide microarchitecture response
  - Early but gentler DTM activation
  - Integrate energy, thermal responses
- *many more…*



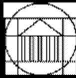
Thermal Modeling: What Next?

- Further validation
  - Direct physical measurements from a real processor & workload would be ideal
  - In the meantime, further FEM and physical measurements using test chips
- More sophisticated modeling of various packaging options and the effect of heat elsewhere in the computer case
- More accurate modeling of interconnect & clock
- More sophisticated modeling of sensor behavior
- Need to extend the processor thermal model
  - CMP, SMT, other components in system...
  - Lots of other stuff!
- Need ways to automatically model floorplan
- Develop more accurate power projections for future technology nodes
- Benchmarking methodology (currently ad-hoc)



Summary

- Architects can help control thermal effects
- Need proper tools
  - Simple estimation techniques are dangerous
  - Lumped RC model is a computationally efficient and accurate solution
  - Floorplan (thermal diffusion), package, and sensor effects must be included
  - HotSpot is publicly available and useful for system-level studies as well as micro-architectural and even circuits
- DTM can provide runtime cooling and lower manufacturing costs with low performance overhead
  - Localized microarchitectural techniques are promising
- Wide open area, lots of low-hanging fruit!



Ultimate Goal

Integrated, workload-aware, autonomic control of performance, temperature, and energy

Download:

http://lava.cs.virginia.edu/HotSpot

(also tech report with extended results)