

# Power and thermal management

Tajana Šimunić Rosing

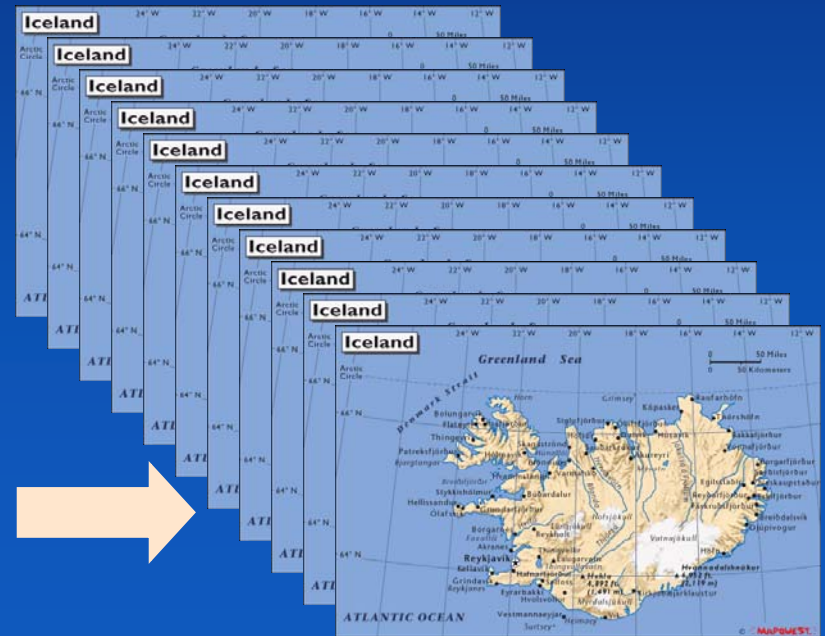
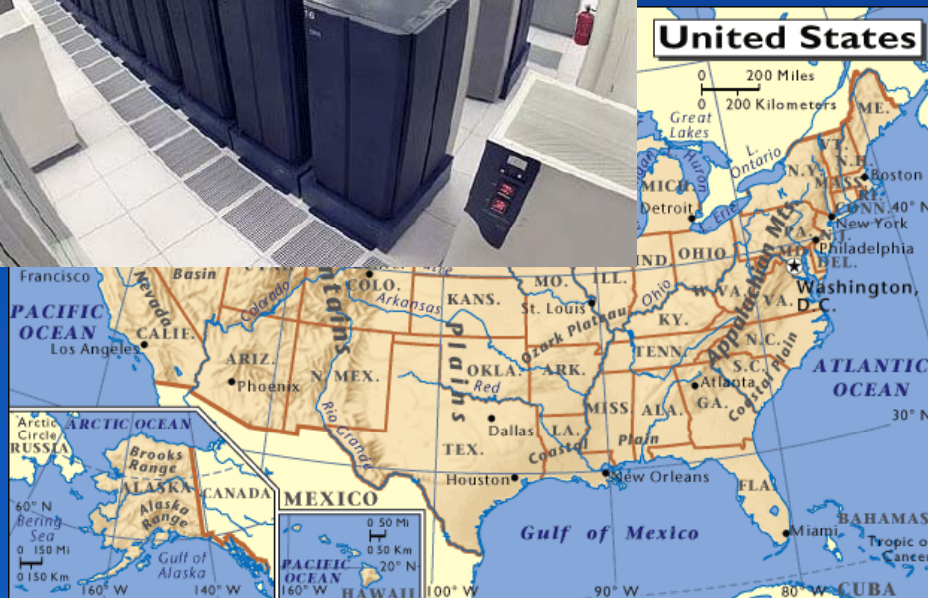
UCSD

# Motivation

- ◆ Power consumption is a critical issue in system design today
  - ❖ Mobile systems want maximum battery lifetime
  - ❖ High performance systems need to reduce the electricity costs
    - Power and cooling



Electricity cost devoted to powering and cooling  
USA data centers can power **10 Icelands!**



# Power and Thermal Management

## ◆ Reducing power $\neq$ lower thermal density

## ◆ Power management

- ❖ Sleep states – DPM
- ❖ Performance states – DVFS

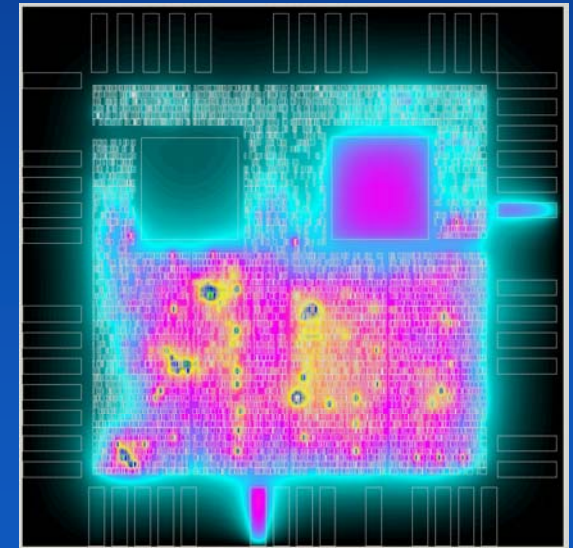
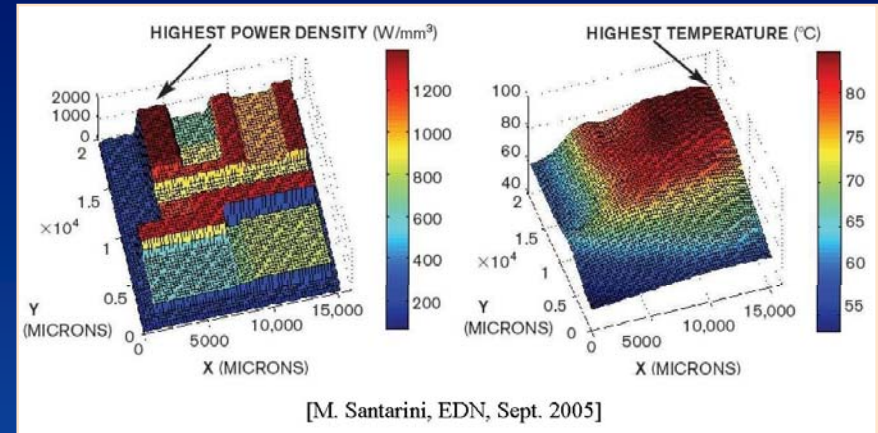
## ◆ Thermal management

### ❖ Thermal Hot Spots

- High leakage power
- Degraded reliability
- Increased interconnect resistivity

### ❖ Spatial and Temporal Gradients

- Higher permanent failure rates
- Timing failures
- Increased interconnect delay and IR drop



# Our Recent Work

## Dynamic power management (DPM)

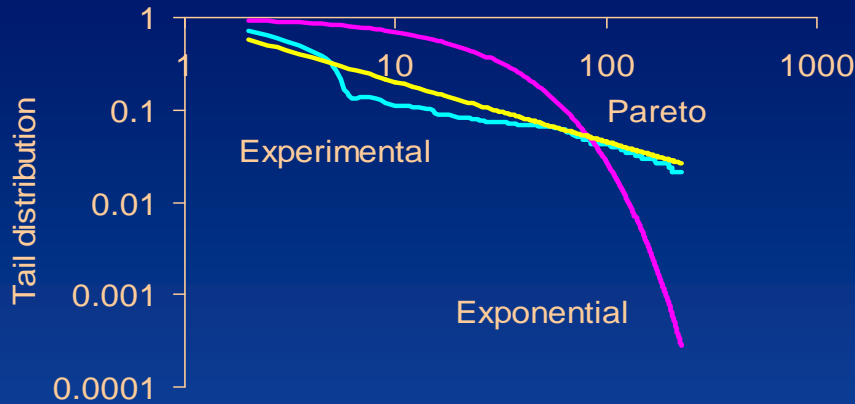
- Optimal power management for stationary workloads
- Machine learning to adapt in non-stationary environments
  - Select among specialized policies
  - Use hardware performance counters to adapt voltage/frequency settings at run time
- Measured large power savings in real systems

## Dynamic thermal management (DTM)

- Workload scheduling:
  - Comparison between power only and thermal management
  - Runtime adaptation to get best temporal and spatial profiles
  - Negligible performance overhead
- Accurate run-time temperature estimation
  - Limited number of thermal sensors in suboptimal locations

# DPM: Workload modeling - Idle State

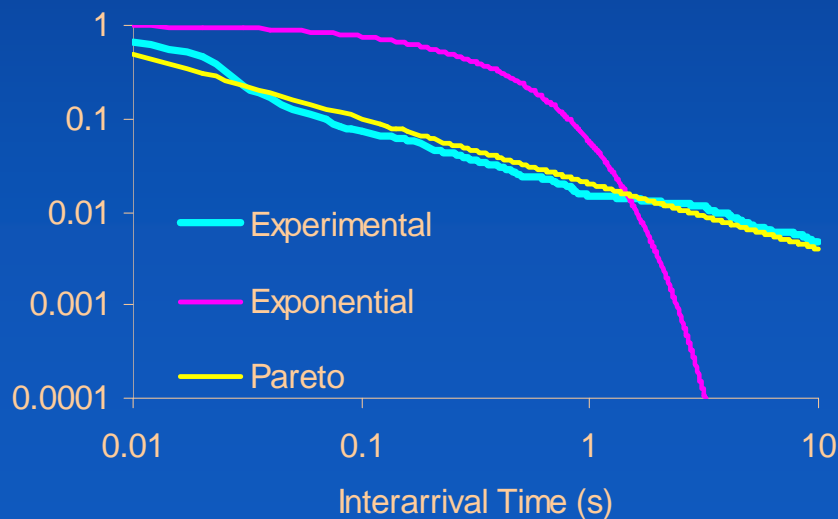
## Hard Disk Trace



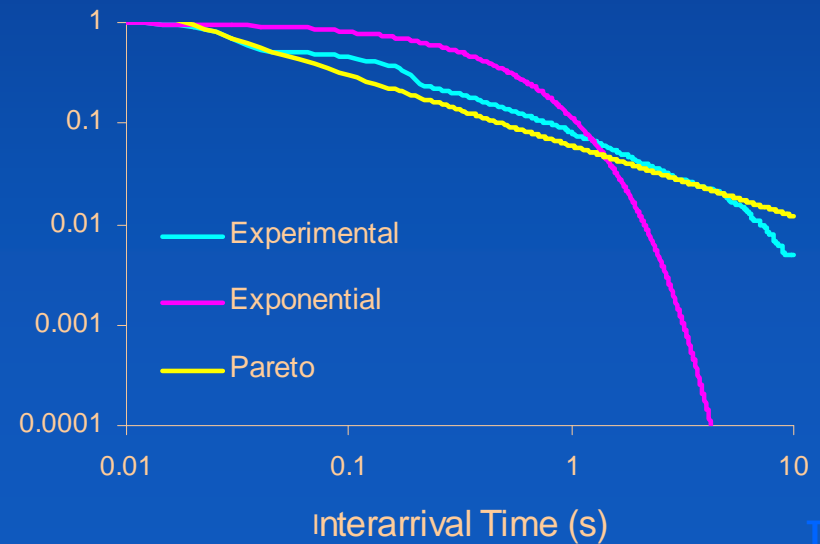
**Pareto Distribution:**

$$E_{user} = 1 - a \cdot t^{-b}$$

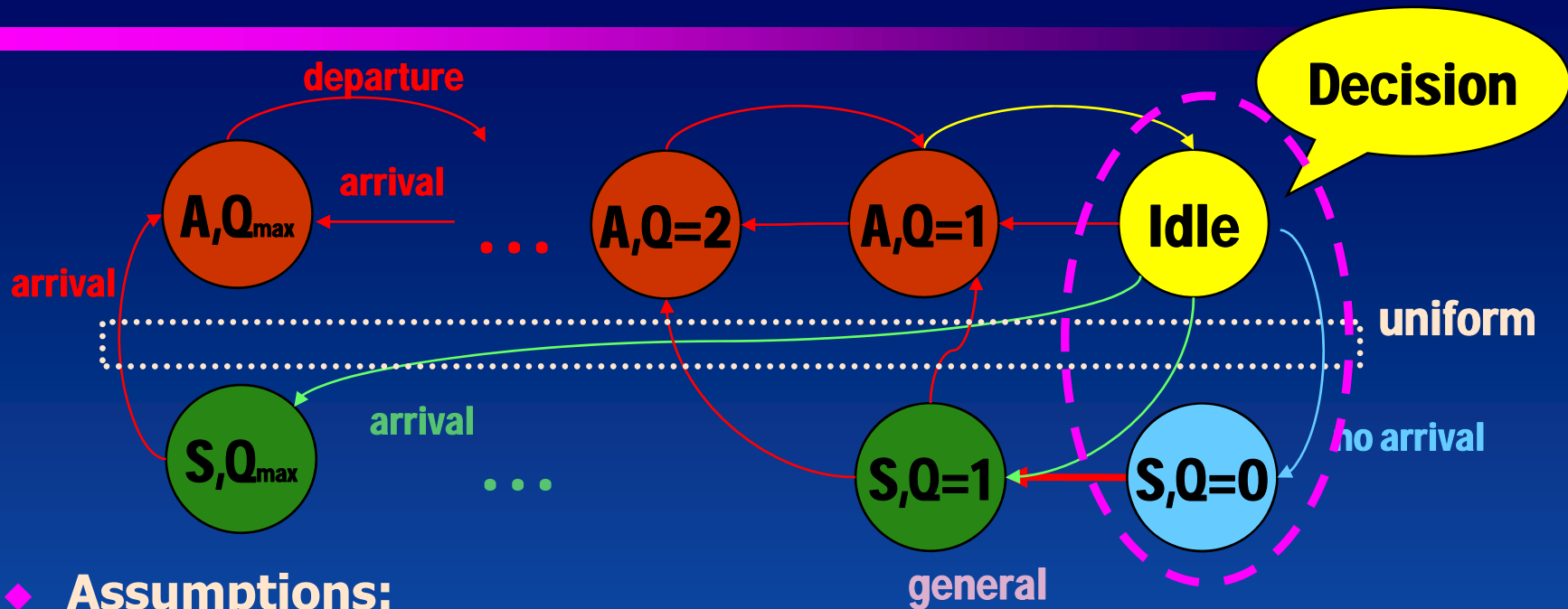
## WWW Trace



## Telnet Trace



# DPM: TISMDP model



## ◆ Assumptions:

- ❖ general distribution governs the first request arrival
- ❖ exponential distribution represents arrivals after the first arrival
- ❖ user, device and queue are stationary

**Obtain globally optimal policy using linear programming**

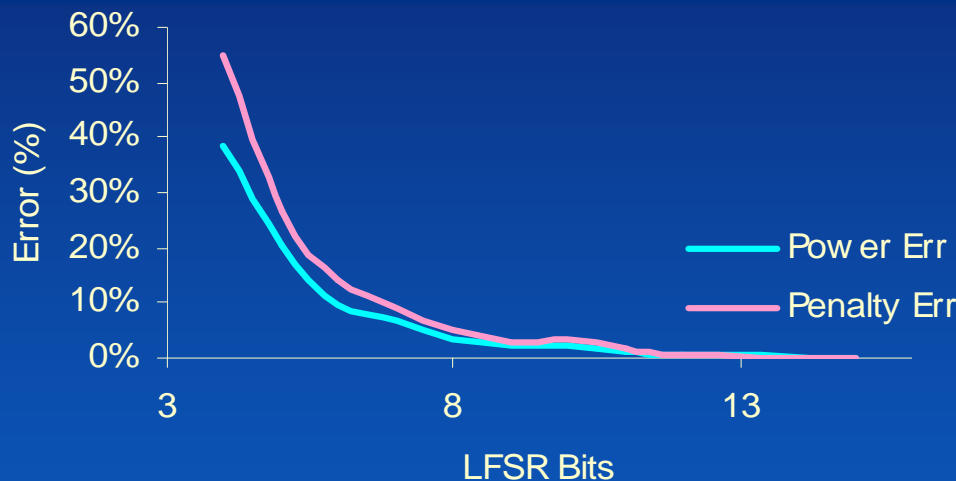
Measurements on hard disk within **11%** of ideal oracle policy  
**factor of 2.4** lower than always-on  
**factor of 1.7** lower than default time-out

# DPM: Hardware implementation

- ◆ LFSR for generating probability & policy logic
- ◆ Controller on entry to idle state:
  - ❖ obtains a random number RND & finds a timeout value (jh) for which  $RND > p(jh)$
  - ❖ if no arrival during jh seconds, the core enters sleep state, otherwise it stays active

## Optimal Policy

Idle time (ms)	Probability to sleep
jh	p(jh)
0	0.00
10	0.00
20	0.12
30	0.43
40	0.75
50	0.87
60	0.91
70	1.00



## FPGA synthesis

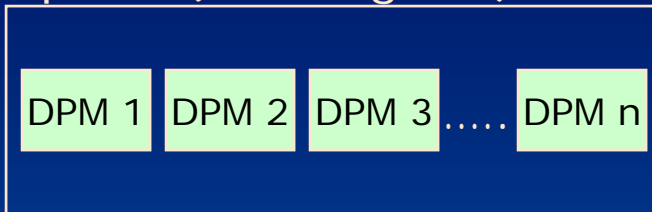
LFSR	LFSR Regs		Policy	
	# LABs	Max ns	# LABs	Max ns
5-15	1	4	2	35

## Synposys synthesis

LFSR Regs		Policy	
#FFs	% area	#gates	% area
5	14%	193	86%
9	14%	417	86%
15	12%	855	87%

# DPM: Handling non-stationary workloads - Machine Learning for DPM

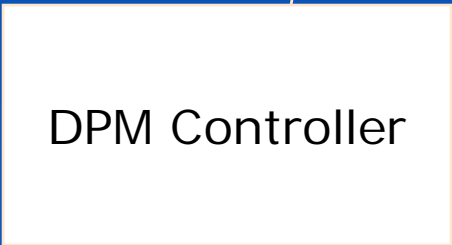
DPM/DVS Experts (Working Set)



Selected expert manages power for the idle period

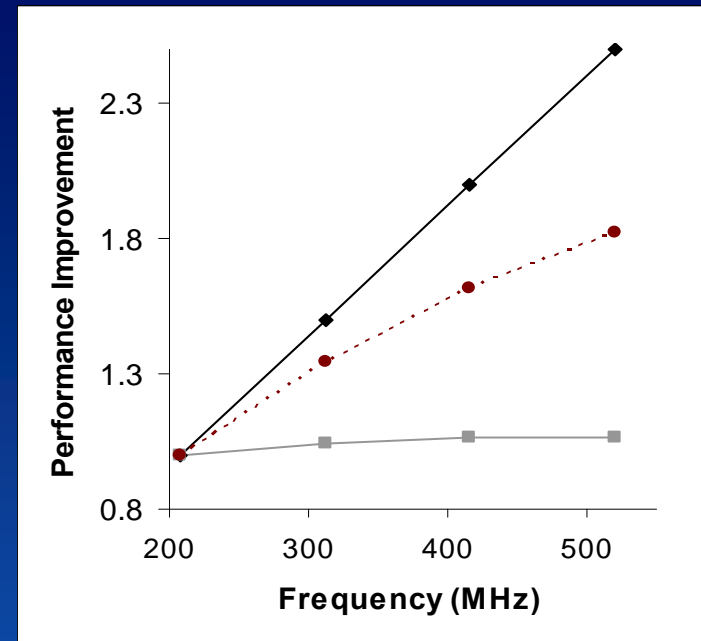
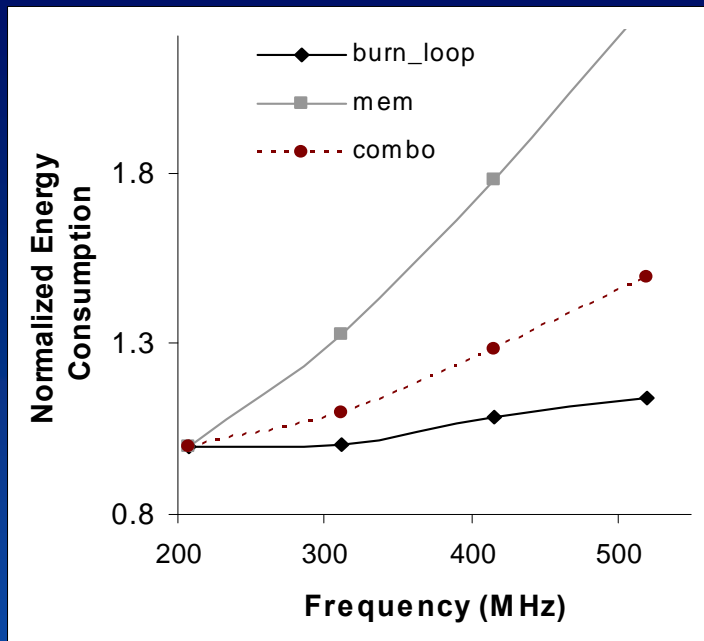


Selects the best performing expert for managing power



Evaluates performance of all experts for that idle period

# Workload Characterization & V/f Selection



- ◆ Three tasks *burn\_loop* (CPU-intensive), *mem* (memory intensive) and *combo* (mix) run with static scaling.
  - ❖ *burn\_loop* has nearly constant energy consumption
  - ❖ *mem* energy efficient at lowest v-f setting
- ◆ Key observation:
  - ❖ CPU-intensive tasks don't benefit from scaling
  - ❖ Memory intensive tasks energy efficient at low v-f settings

# DPM/DVFS: Controller Algorithm

Do for  $t = 1, 2, 3, \dots, T$

1. Calculate  $\mu = \text{CPI}_{\text{base}} / \text{CPI}_{\text{avg}}$

2. Update weight vector of task:

$$w_i^{t+1} = w_i^t \cdot [1 - (1 - \beta) \cdot \text{loss}_i^t(\mu)]$$


3. Choose expert (1, 2, 3...N) with highest probability factor in  $r^t$ :

$$r^t = \frac{w^t}{\sum_{i=1}^N w_i^t}$$

4. Apply the v-f/DPM settings.

5. Reset and restart the Perf. Monitoring Unit

Scheduler tick or idle period start



$$\text{CPI}_{\text{avg}} = \text{CPI}_{\text{base}} + \text{CPI}_{\text{cache}} + \text{CPI}_{\text{tlb}} + \text{CPI}_{\text{branch}} + \text{CPI}_{\text{stall}}$$

Performance converges to that of the best performing expert with successive idle periods at rate  $O\left(\sqrt{(\ln N) / T}\right)$

# Policies used in experiments

- ◆ Hard disk drive

Expert	Characteristics
Fixed Timeout	Timeout = $7 * T_{be}$
Adaptive Timeout	Initial timeout = $7 * T_{be}$ ; Adjustment = $+0.1T_{be} / -0.1T_{be}$
Exponential Predictive	$I_{n+l} = a i_n + (1 - a) . I_n$ , with $a = 0.5$
TISMDP	Optimized for delay constraint of 3.5% on HP-1 trace

Trace Name	Duration (in sec)	$\overline{t_{RI}}$	$\sigma_{t_{RI}}$
HP-1Trace	32311	20.5	29
HP-2 Trace	35375	5.9	8.4
HP-3 Trace	29994	17.2	2

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

- ◆ CPU: Xscale

- ◆ Workloads:

- ◆ qsort, djpeg, blowfish, dgzip

Freq (MHz)	Voltage (V)
208	1.2
312	1.3
416	1.4
520	1.5

# HDD results: Perf Delay/Energy Saving

## With Individual Experts

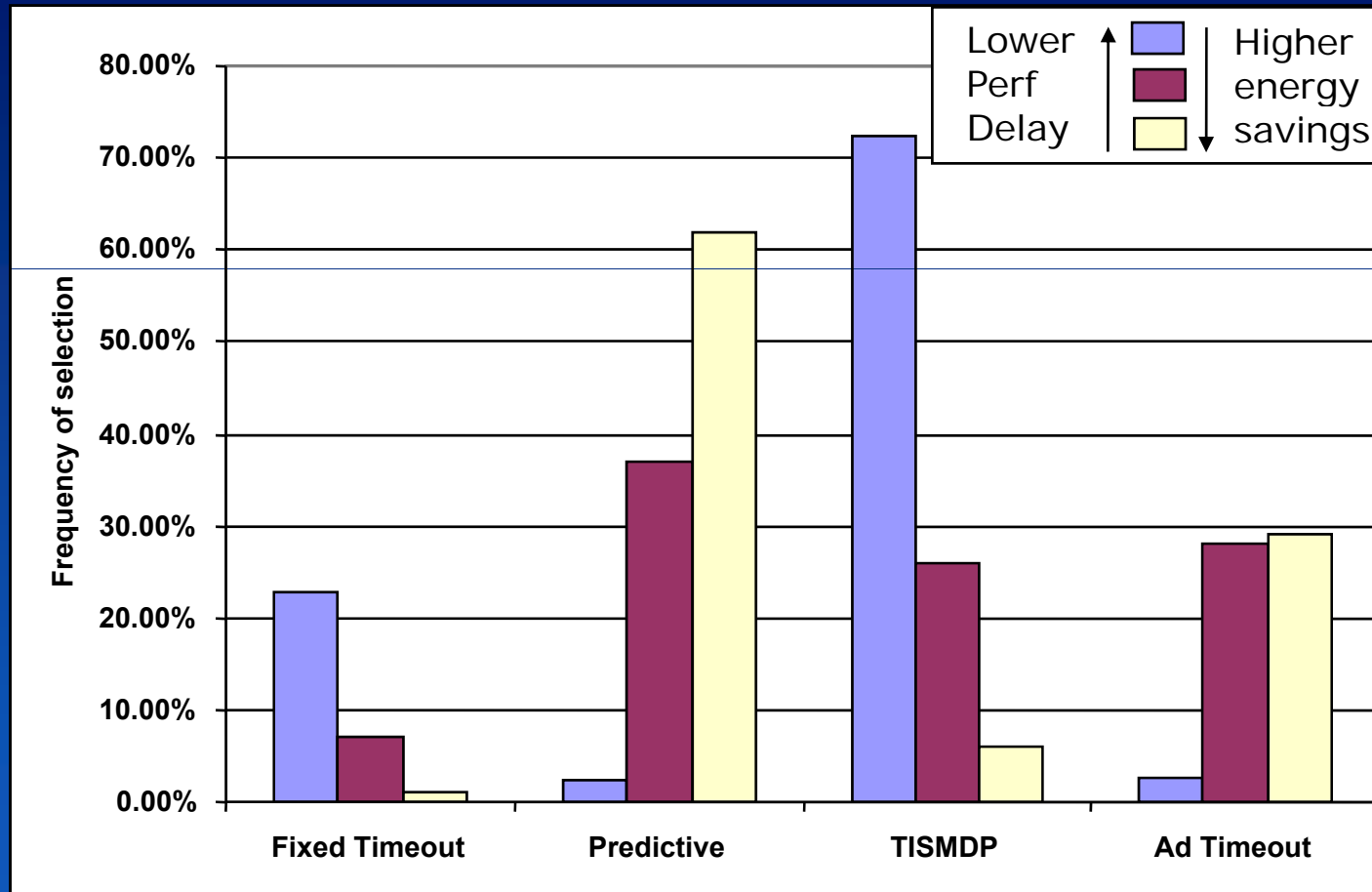
Policy	HP1 Trace		HP2 Trace		HP3 Trace	
	%delay	%energy	%delay	%energy	%delay	%energy
Oracle	0	68.17	0	65.9	0	71.2
Timeout	4.2	49.9	4.4	46.9	3.3	55
Ad Timeout	7.7	66.3	8.7	64.7	6	67.7
TISMDP	3.5	44.8	2.26	36.7	1.8	42.3
Predictive	8	66.6	9.2	65.2	6.5	68

Converges to Predictive

## Energy With Controller

Preferences	HP-1 Trace		HP-2 Trace		HP-3 Trace	
	%delay	%energy	%delay	%energy	%delay	%energy
Low delay	3.5	45	2.61	37.41	2.55	49.5
↓	6.13	60.64	5.86	54.2	4.36	61.02
High energy savings	7.68	65.5	8.59	64.1	5.69	66.28

# HDD results: Frequency of Selection



# DVS: Single Task Environment

- ◆ Single task environment – energy savings up to 50%

Bench.	Low perf delay -----> Higher energy savings					
	%delay	%energy	%delay	%energy	%delay	%energy
qsort	6	17	16	32	25	41
djpeg	7	21	15	37	26	45
dgzip	15	30	21	42	27	49
bf	6	11	16	27	25	40

Bench.	208MHz/1.2V	
	%delay	%energy
qsort	56	48
djpeg	34	54
dgzip	33	54
bf	40	51

- ◆ Multitasking environment – energy savings close to 50%

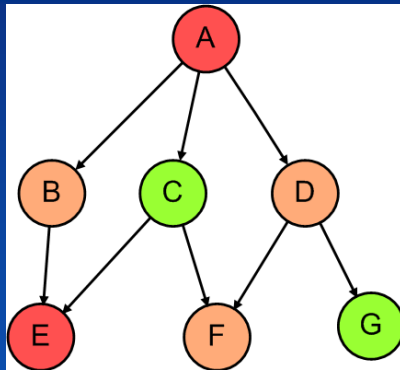
Bench.	Low perf delay -----> Higher energy savings					
	%delay	%energy	%delay	%energy	%delay	%energy
qsort+djpeg	6	17	15	33	25	41
djpeg+dgzip	13	24	19	39	27	48
qsort+djpeg	7	20	18	35	26	42
dgzip+bf	13	18	22	32	27	44

# DTM: Optimal power and thermal thread scheduling

- ◆ Minimize the energy consumption vs. get the optimal temperature distribution

## Workload:

Precedence, timing, thermal characteristics

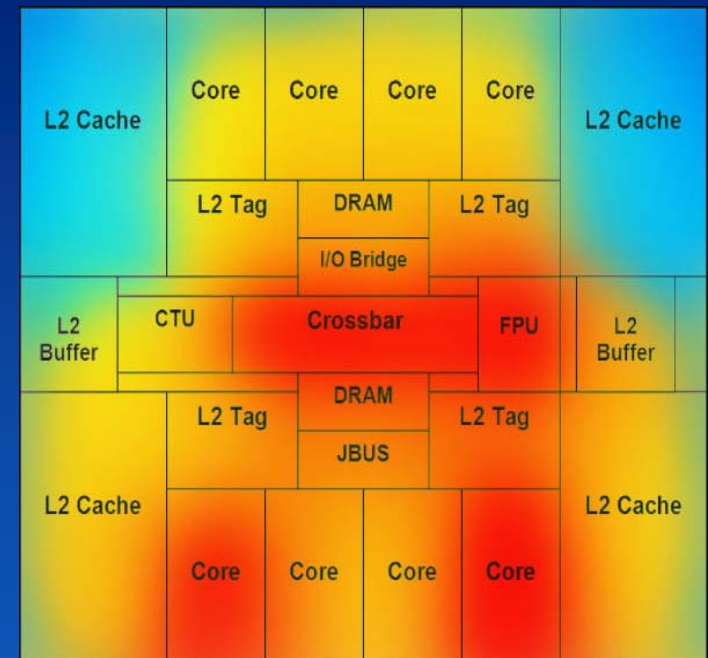


## System Properties:

- Floorplan
- Package

ILP

## Optimal Schedule



# DTM: Evaluation Framework

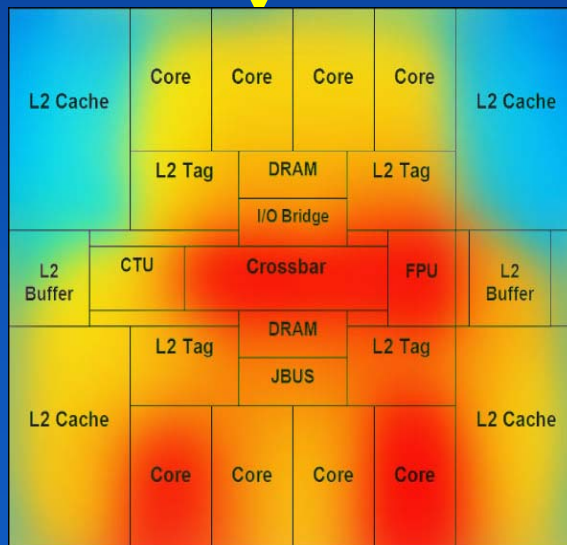
## Inputs:

- Workload information – measured on Niagara
- Floorplan, temperature (for dynamic policies)



## Scheduler

Static: Fixed allocation (ILP)  
Dynamic: Dependent on the policy



Power Manager  
DPM, DVS

## Inputs:

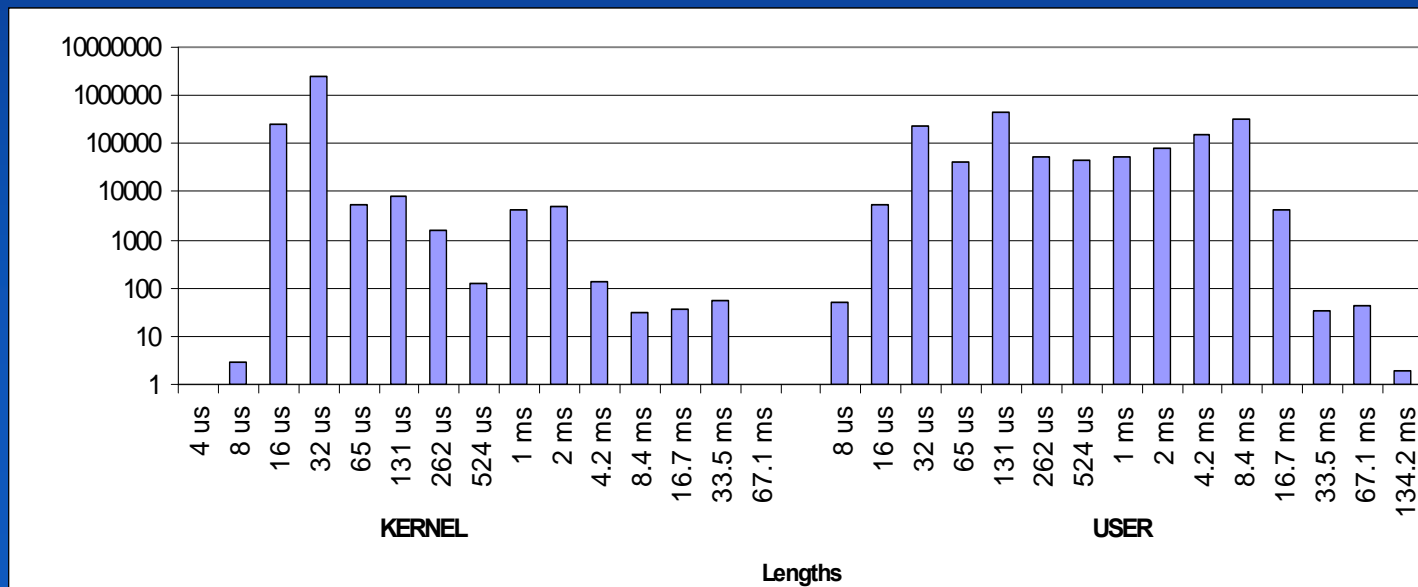
- Power trace for each unit
- Floorplan, package and die properties (Niagara-1)

Thermal Simulator  
HotSpot [Skadron, ISCA'03]

Transient Temp.  
Response for Each Unit

# DTM: Workload

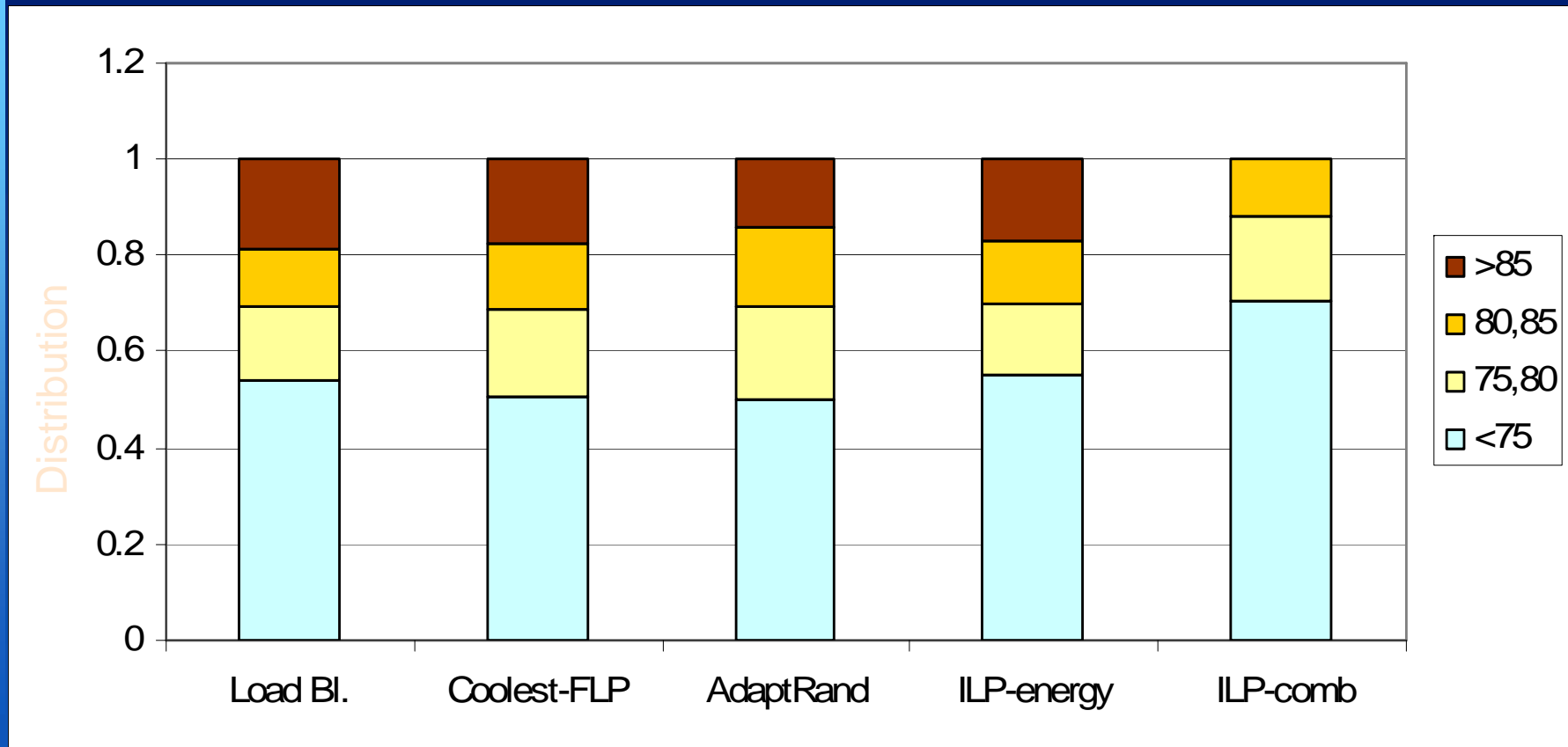
	Utilization (%)			Thread Lengths (ms)		Cache Misses & FP (per 100K instr)			MIPS
	avg	min	max	avg	max	L2 I Miss	L2 D Miss	FP instr	
<b>Web - medium</b>	53.12	28	82	2.7	134	12.9	167.7	31.2	3798
<b>Web - high</b>	95.87	70	100	2.7	268	67.6	288.7	31.2	5264
<b>Database</b>	17.75	0	42	0.4	268	6.5	102.3	5.9	1522
<b>Web &amp; Database</b>	75.12	37	94	0.8	536	21.5	115.3	24.1	4635
<b>INT - gcc</b>	15.25	0	33	7.2	268	31.7	96.2	18.1	1737
<b>INT - gzip</b>	9	0	30	6.3	536	2	57	0.2	1114



# DTM: Policies compared

- ◆ **Optimal and static:**
  - ❖ **ILP-energy**
    - minimizes the overall energy consumption
  - ❖ **ILP-comb**
    - minimizes the thermal hot spots and the temperature gradients
- ◆ **Dynamic:**
  - ❖ **Load balancing**
    - Balances threads for performance only
  - ❖ **Coollest-FLP**
    - Exploits the horizontal heat transfer on the die – schedules threads to cores with “idle” neighbors
  - ❖ **Adaptive-Random Policy**
    - Minimizes & balance temperature with low scheduling complexity
    - *Probability* of sending a workload to a core based on temperature history
    - Adapts to changes in temperature dynamics

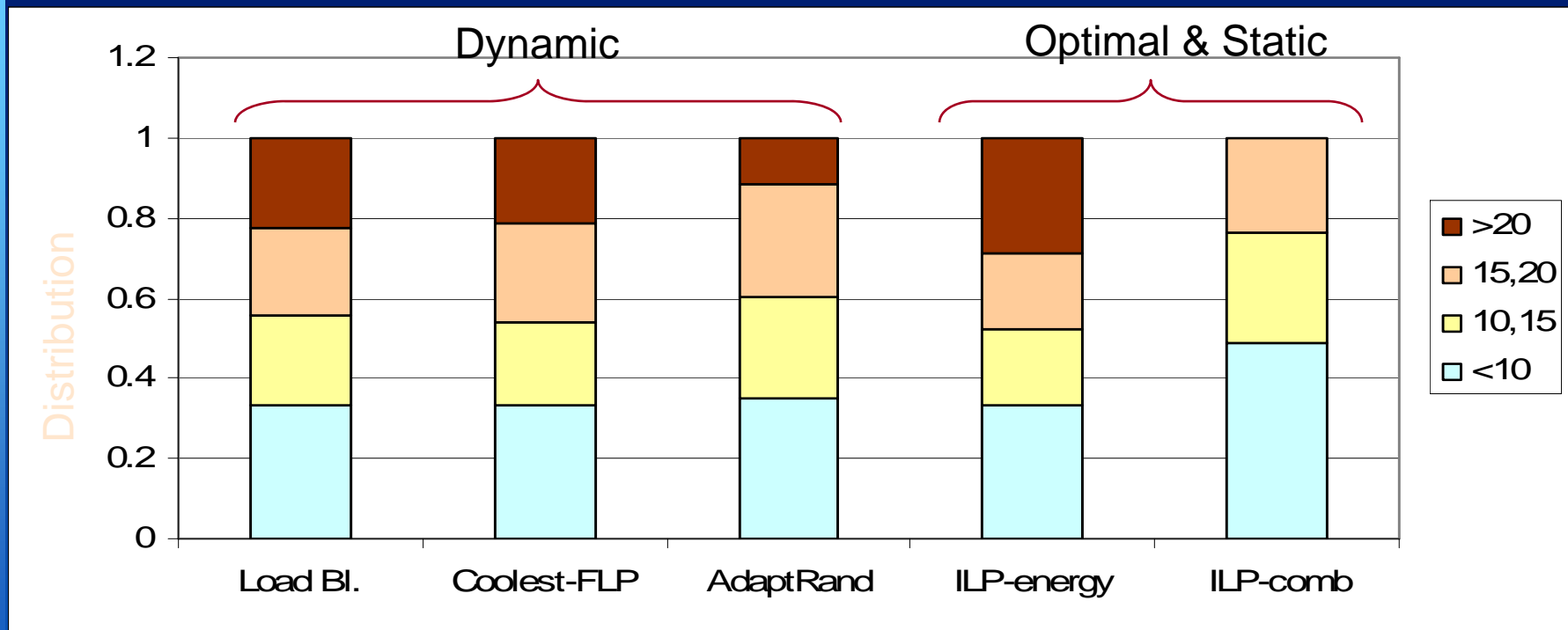
# Results: Thermal Hot Spots



Dynamic

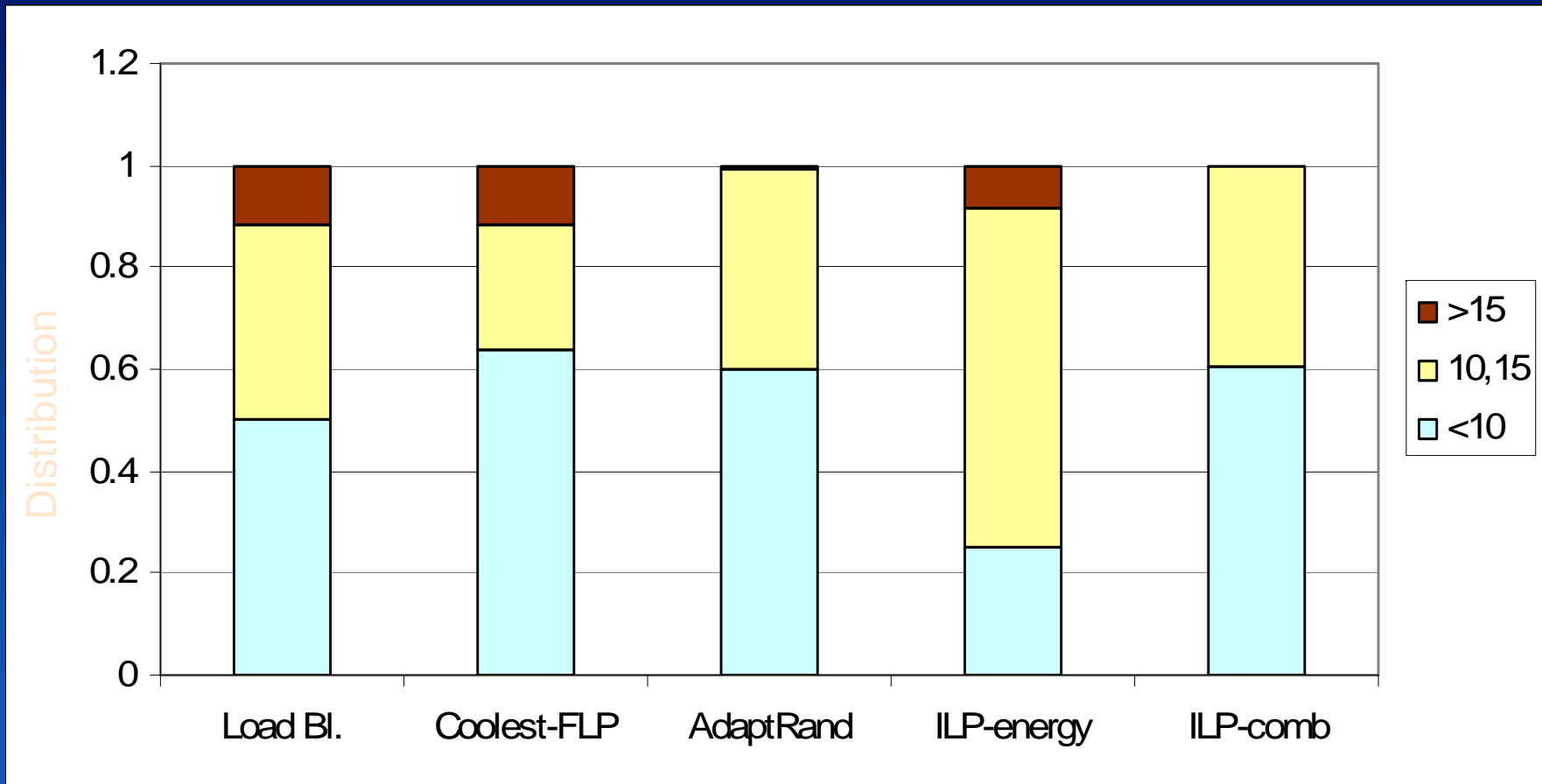
Optimal & static

# DTM: Thermal Cycles



- ◆ Cycles to failure:  $N = C_0 (\Delta T)^{-q}$  ( $q=4$  for metallic structures)
- ◆  $\Delta T$  increases from 10°C to 20°C  
→ Failures happen 16 times sooner

# DTM: Spatial thermal gradients

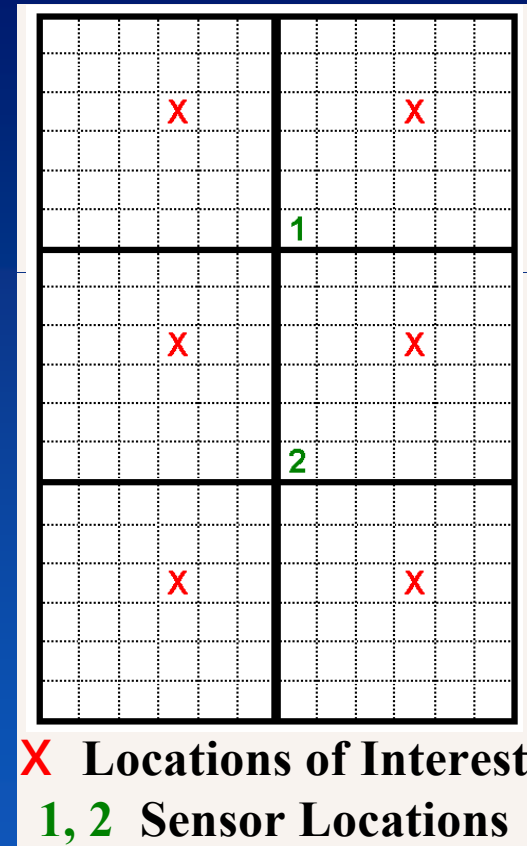


Dynamic

Optimal & Static

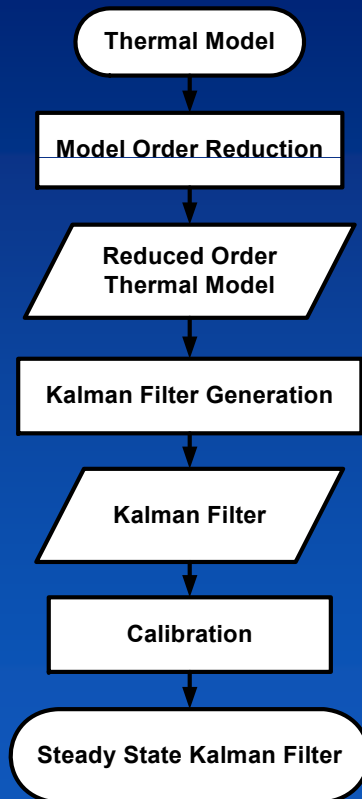
# DTM: Online temperature measurement

- ◆ Limited number of sensors on a device
- ◆ Sensor readings can be inaccurate (noise, calibration, A/D quantization)
- ◆ Accurate temperature estimates are often needed at the points other than sensor locations

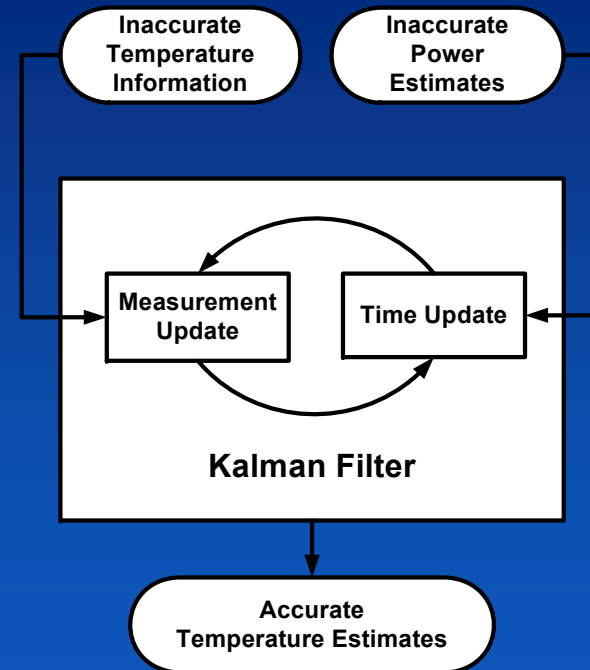


# DTM: Accurate Temperature Estimation

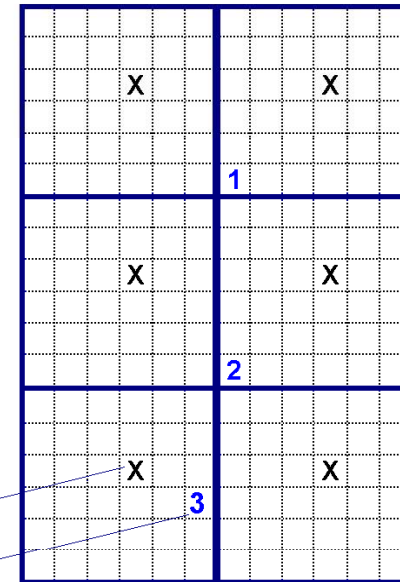
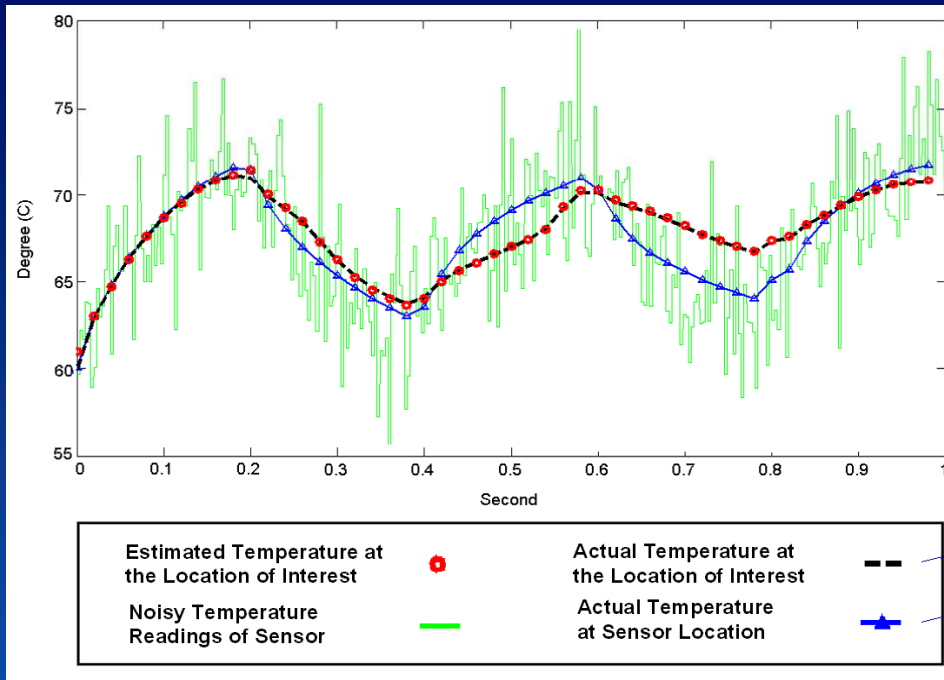
## ◆ Offline phase (Setup)



## ◆ Online phase (Estimation)



# DTM: Results of online temperature estimation



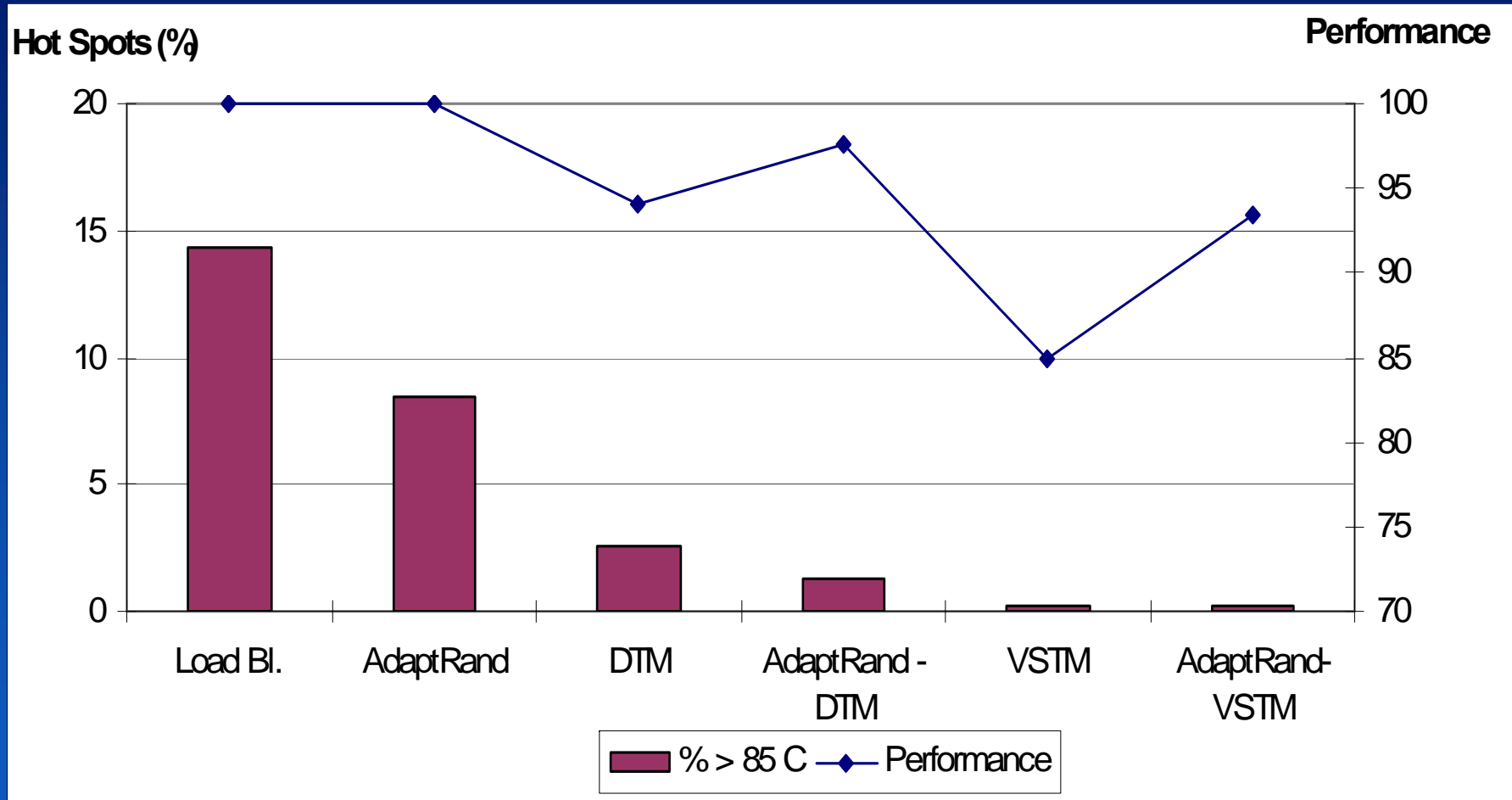
Number of Sensors	Sensor Measurement Errors (°C)		Temperature Estimation Error (°C)	
	Mean Absolute Error	Std. Dev.	Mean Absolute Error	Std. Dev.
2	3.74	4.72	0.77	1.28
3	3.72	4.60	0.76	1.27
4	4.41	5.50	0.75	1.27
5	3.29	3.94	0.76	1.27

- We reduced:
  - Mean absolute temperature error by 5X
  - Standard deviation of the error by 4X

# Conclusions

- ◆ Power management can achieve large energy savings by exploiting variations in workload
  - ❖ TISMDP DPM/DVS policy optimized for stationary workloads
    - ❖ Implementable in hardware
  - ❖ Machine learning to optimally select among individual DPM/DVS policies
- ◆ Minimizing power consumption does not always lead to optimal thermal profiles both in terms of hot spots and temperature gradients
- ◆ Thermal management:
  - ❖ Very low overhead policies minimize hot spots and thermal gradients
  - ❖ Driven by sensors which can be inaccurate
    - ❖ Our temperature estimation uses sensor data to derive accurate thermal profiles online

# Hot spots vs. Performance – Dynamic Techniques



# DPM - Hard Disk Measurement Results

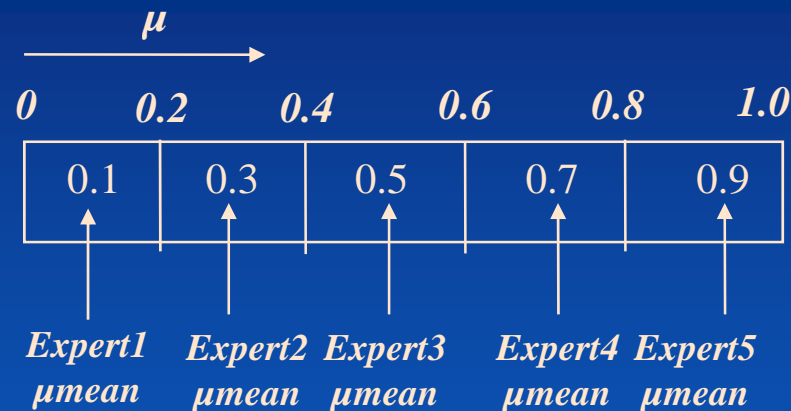
- ◆ Policy is implemented using ACPI standard on the Hard Disk of Sony Vaio laptop running Win NT 5.0 $\beta$
- ◆ measured **real power consumption**
- ◆ 11 hr user trace

Algorithm	Power (W)	T <sub>ss</sub> (s)
Oracle	0.33	118
<b>TISMDP</b>	<b>0.40</b>	<b>81</b>
Karlin's	0.44	79
30s Timeout	0.51	157
<b>120s Timeout</b>	<b>0.67</b>	<b>255</b>
Always on	0.95	0
Poisson	0.97	4

**within 11%** of ideal oracle policy  
**factor of 2.4** lower than always-on  
**factor of 1.7** lower than default time-out

# Evaluation of experts (loss calculation)

- ◆ Intuition: Best suited frequency scales linearly with  $\mu$ .
- ◆ Map task characteristics to the best suited frequency using  $\mu$ -mapper.  
e.g: Experts 1 to 5 = {100,200,300,400,500} MHz
- ◆ Evaluate experts against the best suited frequency.



	Energy Loss ( $l_{ie}^t$ )	Perf Loss ( $l_{ip}^t$ )
$\mu > \mu\text{-mean}$	0	$(\mu - \mu\text{-mean})$
$\mu < \mu\text{-mean}$	$(\mu\text{-mean} - \mu)$	0
Total Loss ( $l_i^t$ ) = $\alpha \cdot l_{ie}^t + (1 - \alpha) \cdot l_{ip}^t$		

$$w_i^{t+1} = w_i^t \cdot (1 - (1 - \beta) \cdot \text{loss}_i^t(\mu))$$

# What about Multi-tasking systems?

- ◆ Tasks with different characteristics can execute together.
- ◆ Weight vector ( $w_t$ ) characterizes an executing task.
- ◆ Need to personalize weight vector at the task level for accurate characterization.
- ◆ Solution: store weight vector as a task level structure



# DVFS: Frequency of Selection

For qsort

