Crank It Up or Dial It Down

Coordinated Multiprocessor Frequency and Folding Control

Augusto Vega, Alper Buyuktosunoglu, Heather Hanson, Pradip Bose, Srinivasan Ramani

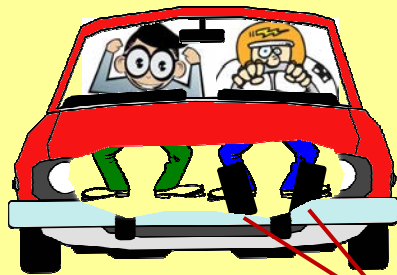**IBM T. J. Watson Research Center, IBM Systems & Technology Group**

Executive Summary

- Modern multi-core systems incorporate support for dynamic power management with multiple actuators

- Algorithms that control these actuators have evolved independently
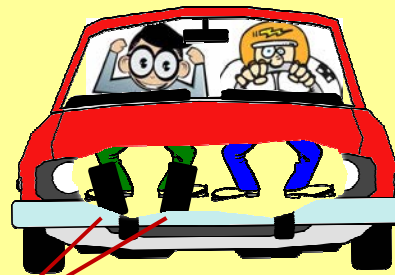 - Their independent operation can result in suboptimal decisions

What is more appropriate?

This?   Or this?

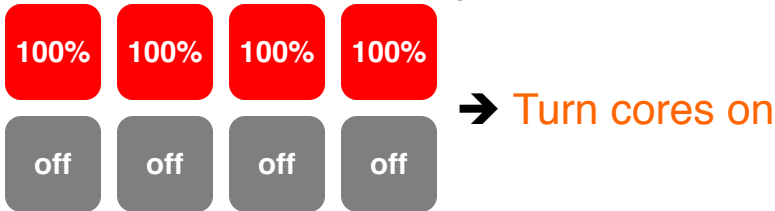Coordinated Control   Pedals   Decoupled Control

We argue in favor of a **coordinated** control of these actuators to avoid potential conflicts in dynamic power management
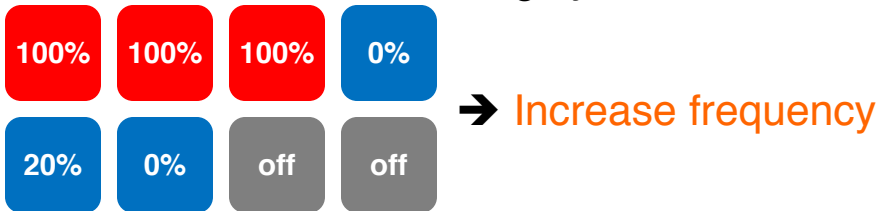
Performance *And* Throughput Awareness

OUR IDEA: Operate power management *knobs* depending on if an application's current execution phase is single-thread performance or throughput bound

All turned-on cores are highly utilized

100%	100%	100%	100%
off	off	off	off

➔ Turn cores on

Some turned-on cores are highly utilized

100%	100%	100%	0%
20%	0%	off	off

➔ Increase frequency

All turned-on cores are low utilized or idle

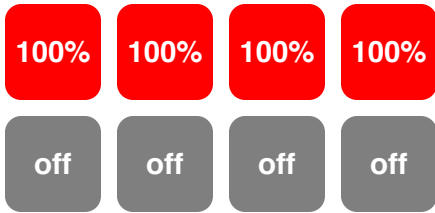| 15%	30%	20%	20%
0%	25%	0%	off

➔ Decrease frequency, turn cores off

Performance *And* Throughput Awareness

OUR IDEA

Operate power management *knobs* depending on if an application's current execution phase is single-thread performance or throughput bound
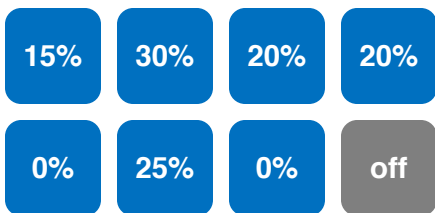
All turned-on cores are highly utilized

| 100% | 100% | 100% | 100% |
| off | off | off | off |

→ Turn cores on

Some turned-on cores are highly utilized

| 100% | 100% | 100% | 0% |
| 20% | 0% | off | off |

→ Increase frequency

All turned-on cores are low utilized or idle

| 15% | 30% | 20% | 20% |
| 0% | 25% | 0% | off |

→ Decrease frequency, turn cores off

PAMPA preserves performance

PAMPA properly actuates DVFS and PCPG

Fluidanimate

- No-PM
- PCPG+DVFS
- PAMPA

Execution Time

System Power

Frequency (GHz)

On Cores

Software Threads: 8, 16, 32, 64, Average