USC

USC Viterbi
School of Engineering

Dynamic MIPS Rate Stabilization in Out-of-Order Processors

Jinho Suh and Michel Dubois

Ming Hsieh Dept of EE
University of Southern California

Outline

USC Viterbi School of Engineering

- **Motivation**
- **Performance Variability of an Out-of-Order Processor**
- **Dynamic MIPS Rate Stabilization**
- **Stabilization Results**
- **Stabilization Framework Robustness**
- **Conclusion**

Motivation

USC Viterbi School of Engineering

- **Timing predictability of a task: top priority**
  - **Dynamic timing analysis, WCET (Worst-Case Execution Time) analysis**
- **WCET analysis : best for hard RT**
  - **But challenging due to advancement in microarchitectures [Hergenhan'00]**
- **Power/energy savings**
  - **Throttling frequency with scheduling upon WCET analysis [Hughes'01] [Rotenberg'01][Zhu'00]**
  - **Exploiting ILP (Instruction Level Parallelism) over frequency [Childers'00]**

Real-time systems avoid Out-of-Order processors with caches

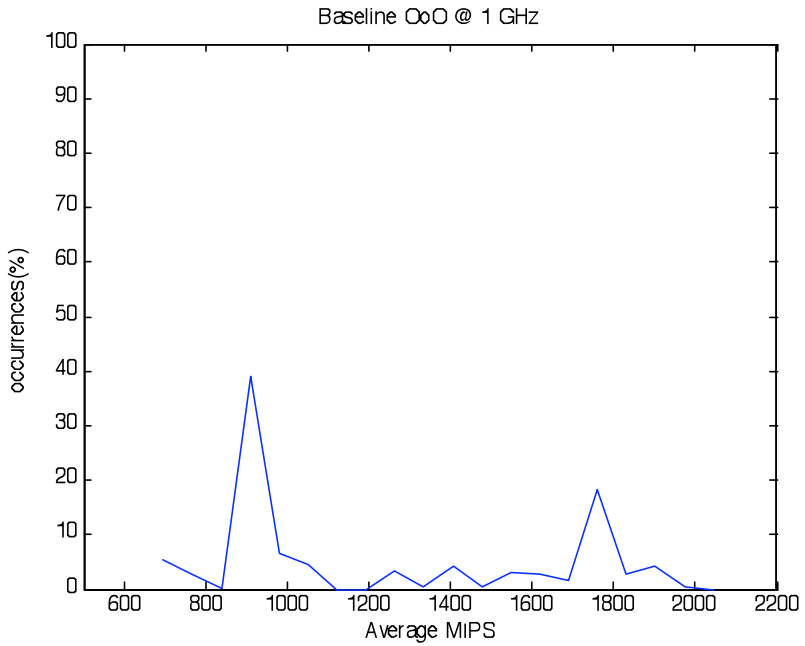
⇩

Stabilization framework: Novel methodology to improve predictability
- **Fine-control of throughput T(maximum instruction count, deadline)**
- **Power and energy savings without task overrun**

Performance Variability of an OoO Processor

USC Viterbi School of Engineering

- **17 MiBench [Guthaus'01] programs**
  - **388 slices (40MI each) : tasks**
- **OoO processor**
  - **@ 1GHz**
  - **3-way**
  - **16KB IL1/DL1, 512KB UL2**
  - **ISA: PISA**
- **Statistics**
  - **Mean: 1307.9 MIPS**
  - **STDEV: 437.6 MIPS**



Timing predictability is difficult problem due to high variability in OoO processors with caches

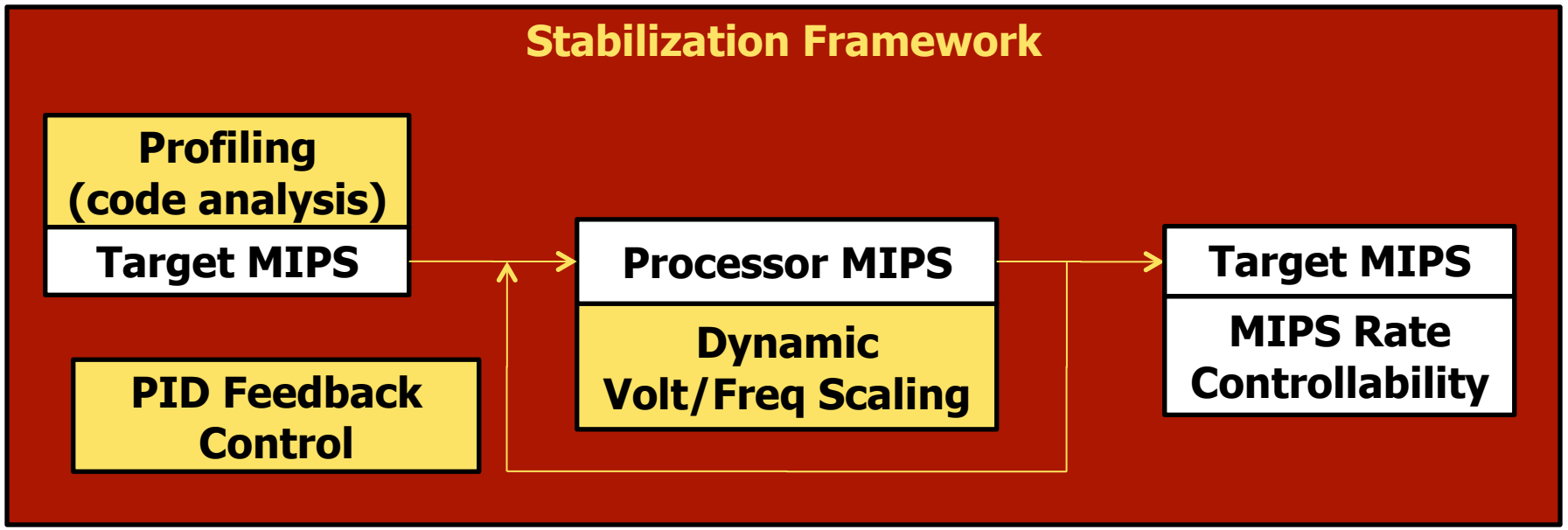
Dynamic MIPS Rate Stabilization

USC Viterbi School of Engineering

Fine Controllability of MIPS Rate

⇩

Just-in-Time Completion without Overrun + **Optimal Power/Energy Savings**



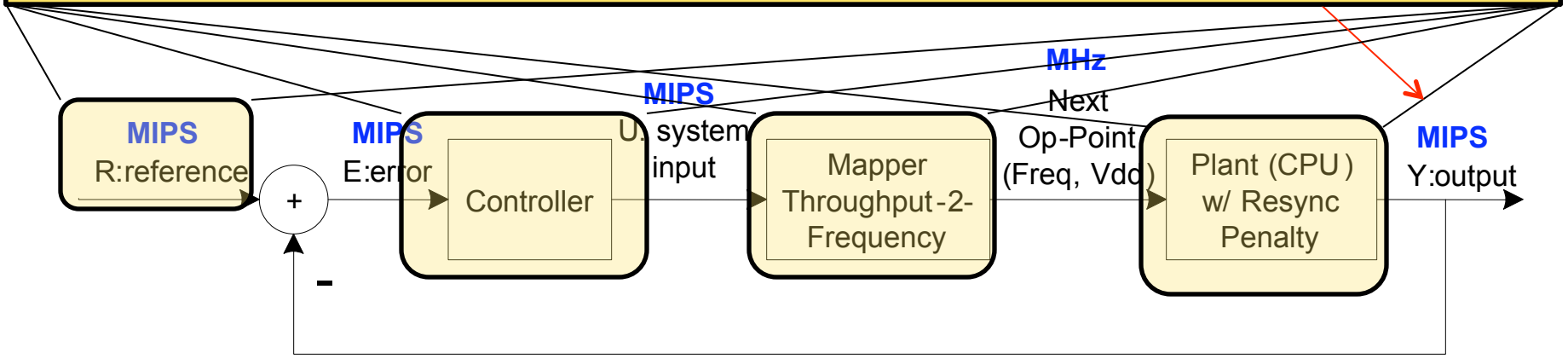
Dynamic MIPS Rate Stabilization

USC Viterbi School of Engineering

MIPS-to-Volt/Freq Mapper

- **MIPS (continuous) ⇒ Frequency (continuous) ⇒ Frequency (discrete)**
- **Assumption: current and next phase behave the same**

$$Next\_Freq = Current\_Freq \times \frac{Next\_MIPS}{Current\_MIPS}$$ **⇒ Proper V/F Scale**

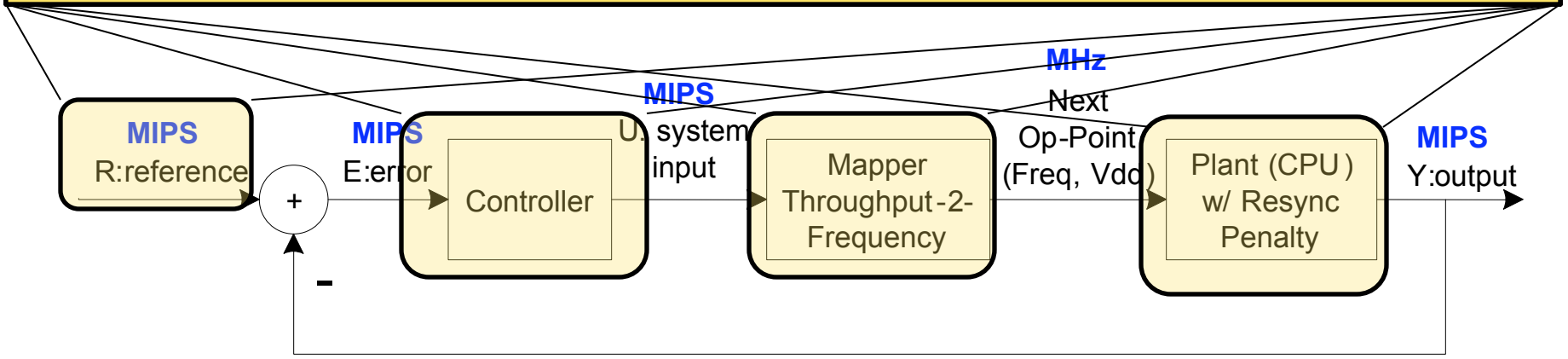


Dynamic MIPS Rate Stabilization: Framework Setup

USC Viterbi School of Engineering

MIPS-to-Volt/Freq Mapper

Next Frequency	Freq (MHz)	Vdd (V)
870MHz ~	1000	0.825
605MHz ~ 870MHz	800	0.772
370MHz ~ 605MHz	500	0.694
~370MHz	300	0.641



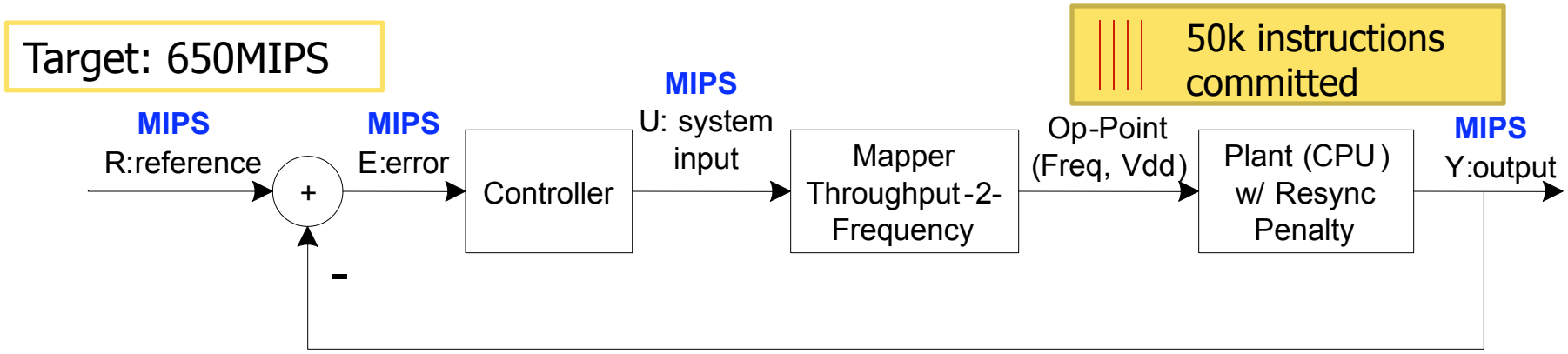
Dynamic MIPS Rate Stabilization

USC Viterbi School of Engineering

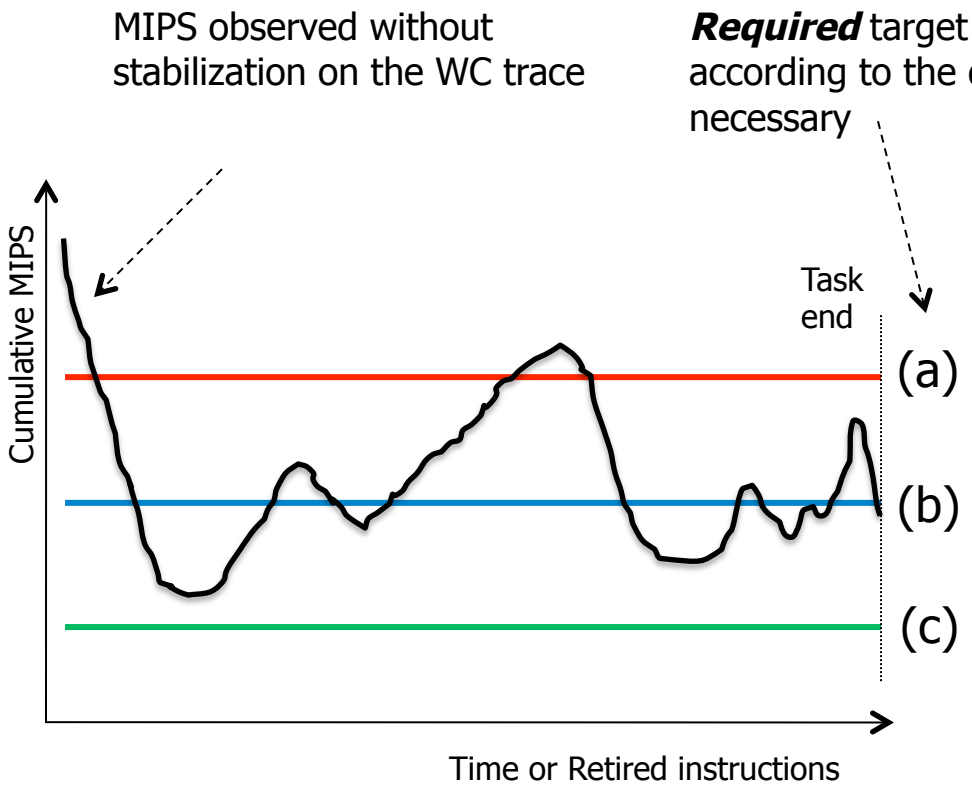
- **Last control window: 656MIPS @ (800MHz, 0.772V)**

- **Current control window: @ (300MHz, 0.641V)**
  - **Measure Current MIPS: 642MIPS**
  - **Calculate Error: 650MIPS – 642MIPS = 8MIPS**
  - **Calculate Next MIPS = 75*(8 – (-6)) + 50*8 = 1450MIPS**
  - **Next Frequency = 300MHz * 1450/642 = 677.57MHz**
  - **Next Volt/Freq = (800MHz, 0.772V)**

- **Next control window:**
  - **PLL Resynchronization pause of 20usec**
  - **Start running @ (800MHz, 0.772V)**



Target MIPS rate and Task Overrun



(a) Target MIPS is not achievable: Task overrun

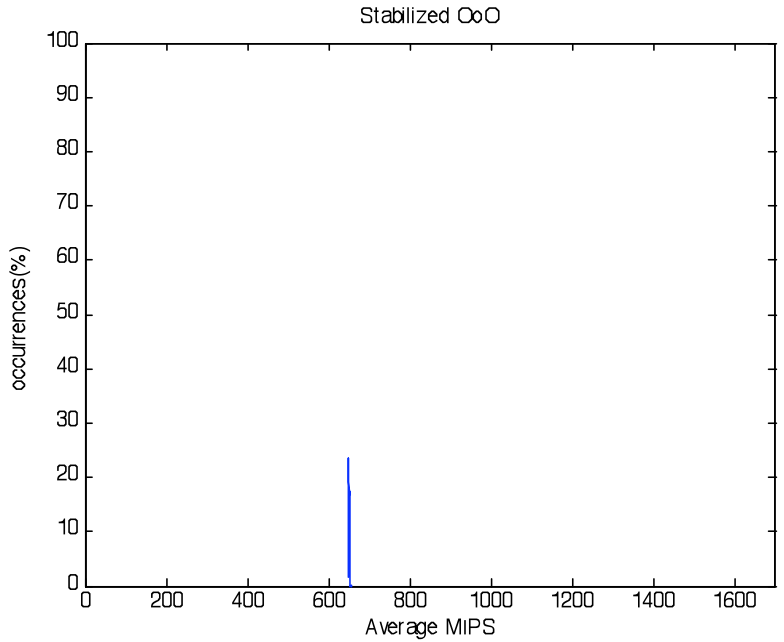
(b) Target MIPS might be achievable: Possible overruns

(c) Target MIPS is achievable: No overrun

Stabilization Results

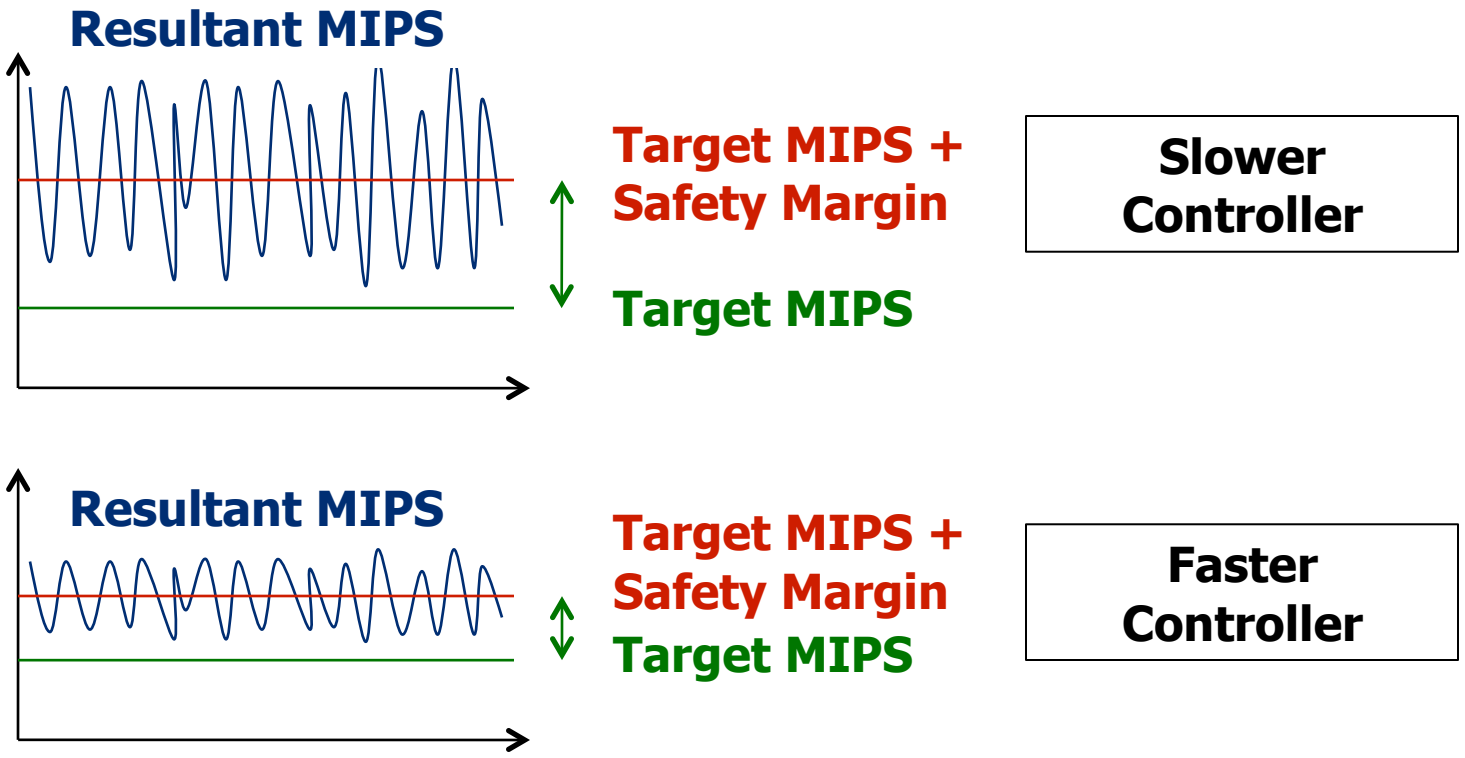
USC Viterbi School of Engineering

- **Target of 650MIPS**
  - **62ms deadline for 40MI task**
- **Statistics**
  - **Mean: 650.37 MIPS**
  - **STDEV: 1.51 MIPS**
- **Savings against baseline**
  - **Average power: 46.64%**
  - **Average energy: 72.06%**



1. **Predictability is much <u>IMPROVED</u> in OoO processors with caches**
2. **Power and energy savings due to just-in-time completion without task overrun**

Stabilization Quality, Safety Margin and Target MIPS



Stabilization Framework Robustness: Different PID parameters

USC Viterbi School of Engineering

PID Parameter Settings	P	I	D	5%	1%	0.5%	0.4%	0.3%	0.1%
Setting 1: Slowest	1	10	0.1						
Setting 2	10	50	0.1						
Setting 3	50	50	0.1						
Setting 4	75	50	0.1			Safe			
Setting 5	100	50	0.1						
Setting 6: Fastest	100	100	0.1						

- **Loop-tuning for PID controller parameters**

From bitcount, qsort, fft_fwr, fft_inv

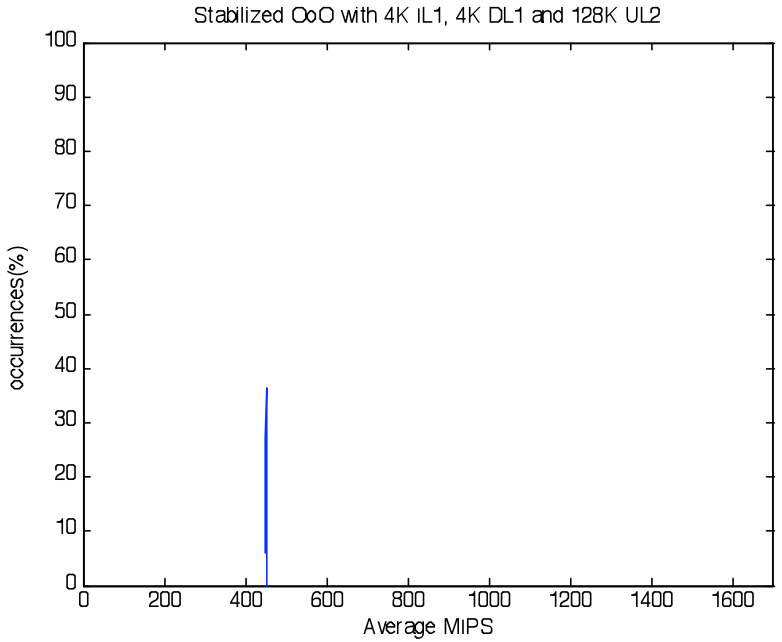
- **No need to *fine tune* parameters**
- **Convergence (settings 2~6): STDEV difference < 1.5MIPS**
- **Power/Energy savings: difference < 10%**

PID controller is ROBUST:
Stabilization works well with different PID parameters

Stabilization Framework Robustness: Different Cache Configurations

USC Viterbi School of Engineering

- **Same PID controller**
- **With 4KB-4KB-128KB caches**
  - **IPC: 62.15%**
  - **IL1 misses: 311.11%**
  - **DL1 misses: 491.86%**
  - **UL2 misses: 609.45%**
- **Statistics:**
  - **Mean: 450.55MIPS**
  - **STDEV: 1.04 MIPS**
- **Same observation with 8KB-8KB-256KB caches**



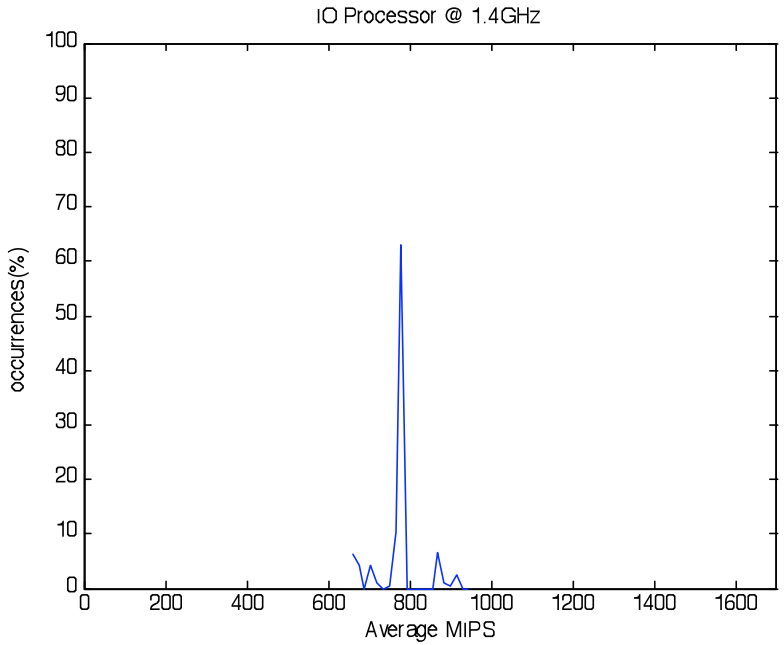
From qsort, patricia

PID controller is <u>ROBUST</u>:
Stabilization works well upon a different cache configurations

Comparison with In-Order (IO) Processor

USC Viterbi School of Engineering

- **Same Quality-of-Service:**
  - **> 650MIPS**
- **IO processor**
  - **@ 1.4GHz**
- **Statistics:**
  - **Mean: 793.14MIPS**
  - **STDEV: 65.5MIPS**
- **Power to stabilized: 151.46%**
- **EPI to stabilized: 110.49%**
- **Power to baseline: 86.19%**
- **EPI to baseline: 102.64%**



From basicmath,, patricia, adpcm_p2a, adpcm_a2p

1. **Stabilized OoO is better than IO for power/energy consumption**
2. **IO can be stabilized as well**

Conclusion

USC Viterbi School of Engineering

- **Fine-grain controllability of processor instruction throughput**
  - **Make execution time highly predictable**
  - **Optimize power/energy consumption by meeting deadlines right on time**
  - **Applicable to many different kinds of (single) processors, including OoO processor with caches for RT applicability**
- **Stabilized OoO processor can be better than IO processor**
- **Robustness**
  - **Over PID parameters, over different cache configurations**
- **Future Work**
  - **Extension of the framework to Chip Multiprocessors**

References

[Cazorla'04] Cazorla, F. J., Knijnenburg, P. M., Sakellariou, R., Fernandez, E., Ramirez, A., and Valero, M. 2004. Predictable performance in SMT processors. In Proceedings of the 1st Conference on Computing Frontiers (Ischia, Italy, April 14 - 16, 2004). CF '04

[Childers'00] Bruce R. Childers, H. Tang and Rami Melhem, Adapting Processor Supply Voltage to Instruction-Level Parallelism, Koolchips 2000, during the 33rd Int'l. Symp. on Microarchitecture (MICRO-33), Monterey, CA, December 10, 2000.

[Burger'97] Doug Burger and Todd M. Austin. The SimpleScalar Tool Set Version 2.0. Technical Report 1342, Computer Sciences Department, University of Wisconsin--Madison, May 1997.

[Guthaus'01] Guthaus, M. R., Ringenberg, J. S., Ernst, D., Austin, T. M., Mudge, T., and Brown, R. B. 2001. MiBench: A free, commercially representative embedded benchmark suite. In Proceedings of the Workload Characterization, 2001. Wwc-4. 2001 IEEE international Workshop on - Volume 00 (December 02 - 02, 2001).

[Hamers'07] Hamers, J. and Eeckhout, L. 2007. Resource prediction for media stream decoding. In Proceedings of the Conference on Design, Automation and Test in Europe (Nice, France, April 16 - 20, 2007). Design, Automation, and Test in Europe. EDA Consortium, San Jose, CA, 594-599.

[Hergenhan'00] A. Hergenhan and W. Rosenstiel. Static timing analysis of embedded software on advanced processor architectures. In Proceedings of Design, Automation and Test in Europe (DATE '00), pages 552--559, Paris, March 2000.

[Hughes'01] C. J. Hughes, J. Srinivasan, and S. V. Adve. Saving Energy with Architectural and Frequency Adaptations for Multimedia Applications. In Proceedings of the 34th Annual International Symposium on Microarchitecture (MICRO-34), Dec. 2001.

[Mistry'04] Mistry, K. Armstrong, M. Auth, C. Cea, S. Coan, T. Ghani, T. Hoffmann, T. Murthy, A. Sandford, J. Shaheed, R. Zawadzki, K. Zhang, K. Thompson, S. Bohr, M. Delaying forever: Uniaxial strained silicon transistors in a 90nm CMOS technology, Symposium on VLSI Technology, p. 50, (2004).

[Rotenberg'01] E. Rotenberg. Using Variable-MHz Microprocessors to Efficiently Handle Uncertainty in Real-Time Systems. 34th International Symposium on Microarchitecture, December 2001.

[Xu'05] C Xu, TM Le, TT Lay, H.264/AVC CODEC: Instruction Level Complexity Analysis. Ninth IASTED International Conference on Internet and Multimedia Systems and Applications; Honolulu, HI; USA; 15-17 Aug. 2005.

[Zhu'00] Zhu, Y. and Mueller, F. Feedback EDF Scheduling Exploiting Dynamic Voltage Scaling. In Proceedings of the 10th IEEE Real-Time and Embedded Technology and Applications Symposium (Rtas'04) - Volume 00 (May 25 - 28, 2004).