Evaluating and Exploiting Impacts of Dynamic Power Management Schemes on System Reliability

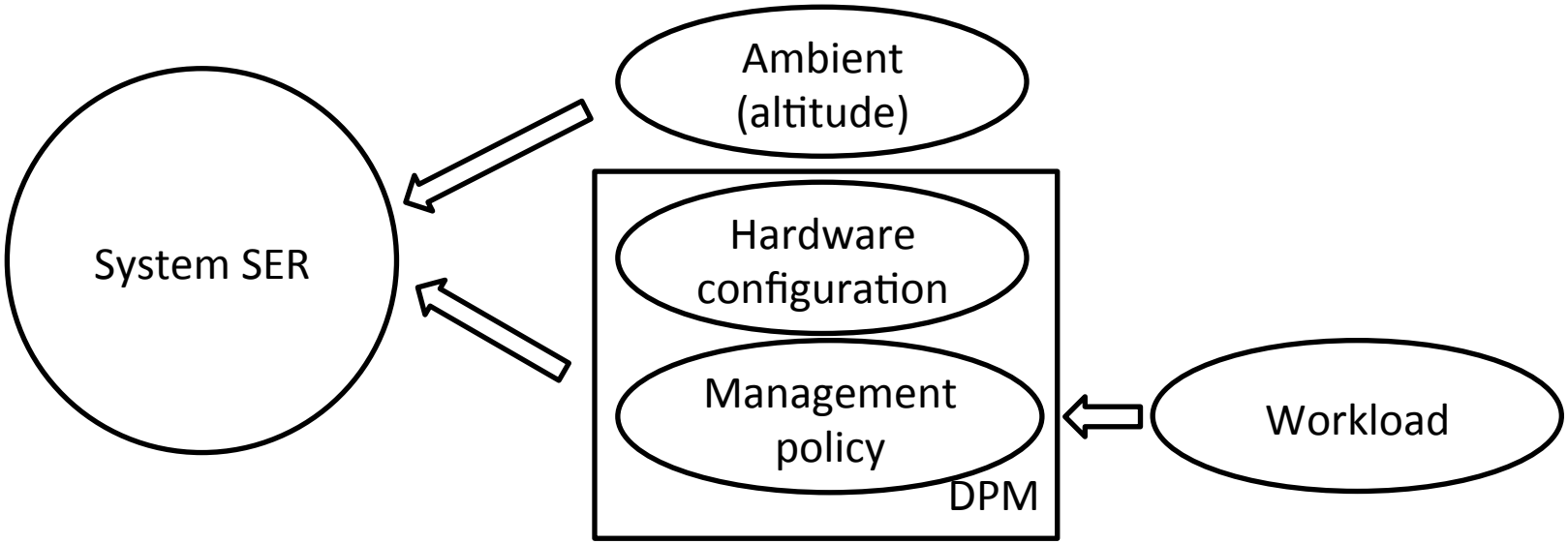
Liangzhen Lai, Vikas Chandra* and Puneet Gupta
UCLA Electrical Engineering Department
ARM Research*

Radiation-Induced Soft Error

- Caused by Alpha particles or neutrons
  - Results in flipping logic states or glitches
- Memory
  - Well-protected if with ECC
- Flip-Flops (FFs)
  - Expensive to fix
- Soft error rate (SER) is measured by failures in time (FIT)
  - Failures per billion device-hours

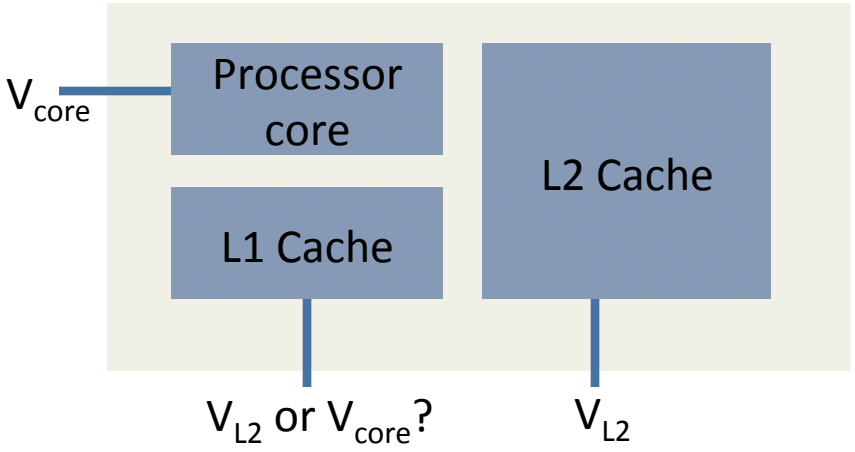
What Can Affect System Reliability?

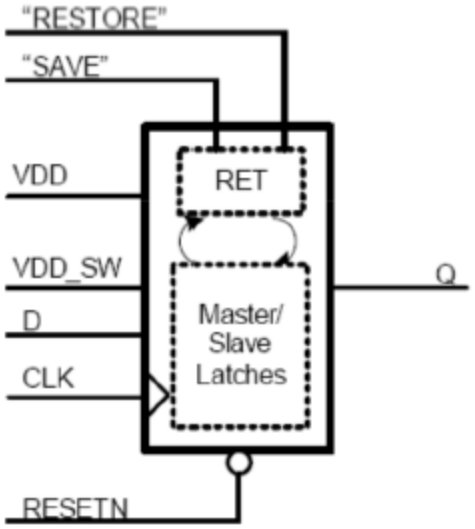
- Component FIT rate depends on:
  - Circuit properties, e.g., schematic, layout
  - Operating conditions, i.e., voltage, location/altitude
- System FIT rate can be affected by:
  - Ambient
  - Dynamic power management (DPM)



Hardware Support for Dynamic Power Management

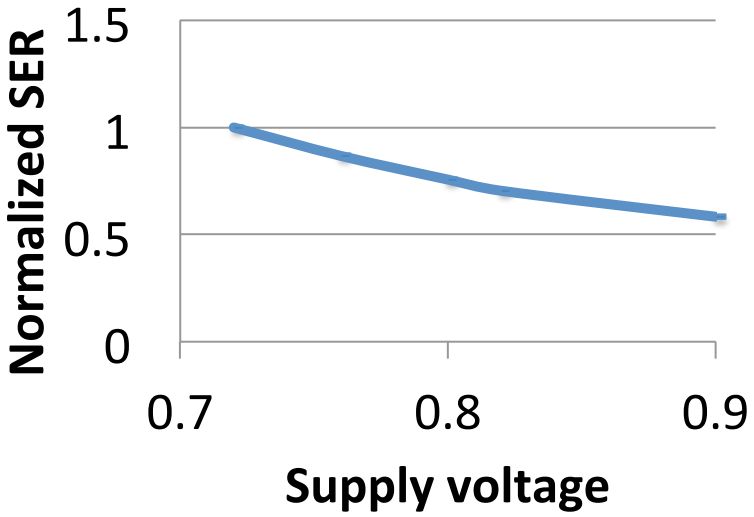
- Voltage scaling
  - L1 cache tie to V_{core}
    - Higher performance
  - L1 cache tie to V_{L2}
    - More power saving
- Power gating
  - Dumping states to memory
    - Simpler
  - Use state-retention FF
    - Faster

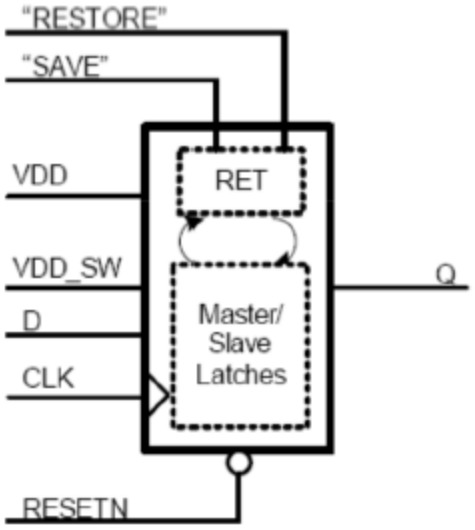




Impacts of Hardware Power States

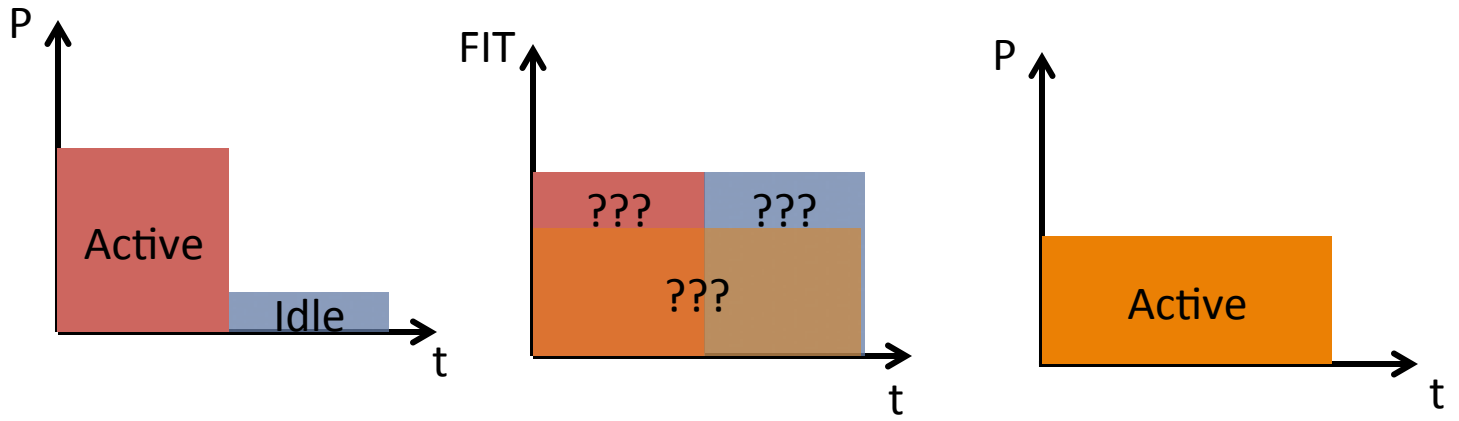
- Voltage states
  - Affects core SER
  - Affects mem SER if L1 is tied to the core
- Sleep states
  - Mem SER
    - Increase if SRAM goes to retention mode
  - Core SER
    - Zero if power gated
    - Increase if using retention FFs





JIT vs RFTS

- Two power management approaches:
  - Run-Fast-Then-Stop (RFTS): complete the workload with nominal performance and then go to certain sleep states for the remaining slack to reduce power
  - Just-In-Time (JIT): adjust the peak performance to elongate the runtime with lower power consumption



Outline

- Motivation
- Evaluating DPM on System Reliability
  - Evaluation Platform
  - Results
- Exploiting DPM for System Reliability
  - Virtual SER monitor
  - Forbidden state based policy
  - Results
- Conclusion

Evaluation Platform Choices

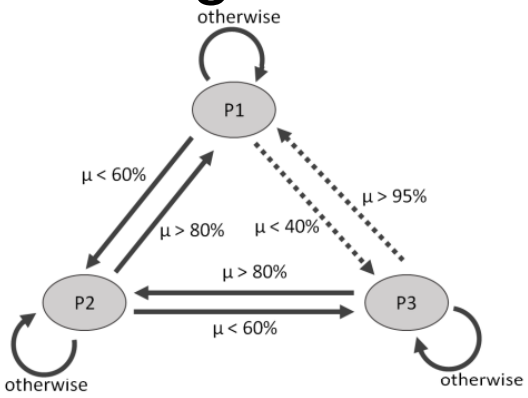
- Running system software (i.e., OS)
  - Simulators are too slow
- Interaction with power management policies
  - Need accurate timing models
- Targeting different hardware configuration
  - Designing hardware for each configuration is infeasible
- Using hardware development board
  - Running applications with OS
  - Emulating the power/SER

Reliability Evaluation

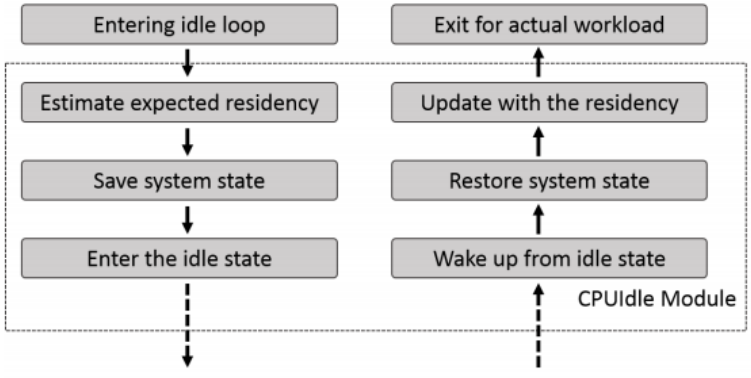
- Hardware:

	Retention FF	L1 Rail	L1 ECC
Case I	No	Same as core	No
Case II	No	Separate	No
Case III	Yes	Same as core	No
Case IV	No	Same as core	Yes

- Software power management schemes
  - Linux governors

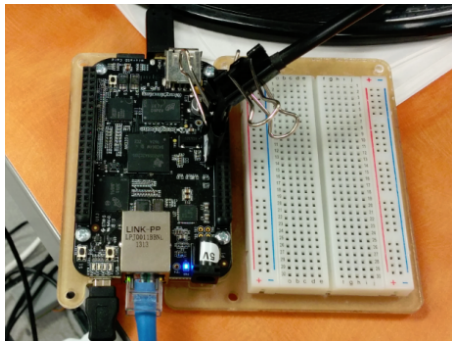


CPUFreq



CPUIdle

Building An Evaluation Platform



- BeagleBone Black
  - Cortex-A8
  - Standard Linux kernel 3.12
  - Supports both CPUFreq and CPUIdle
  - Power/SER models based on 45nm technology

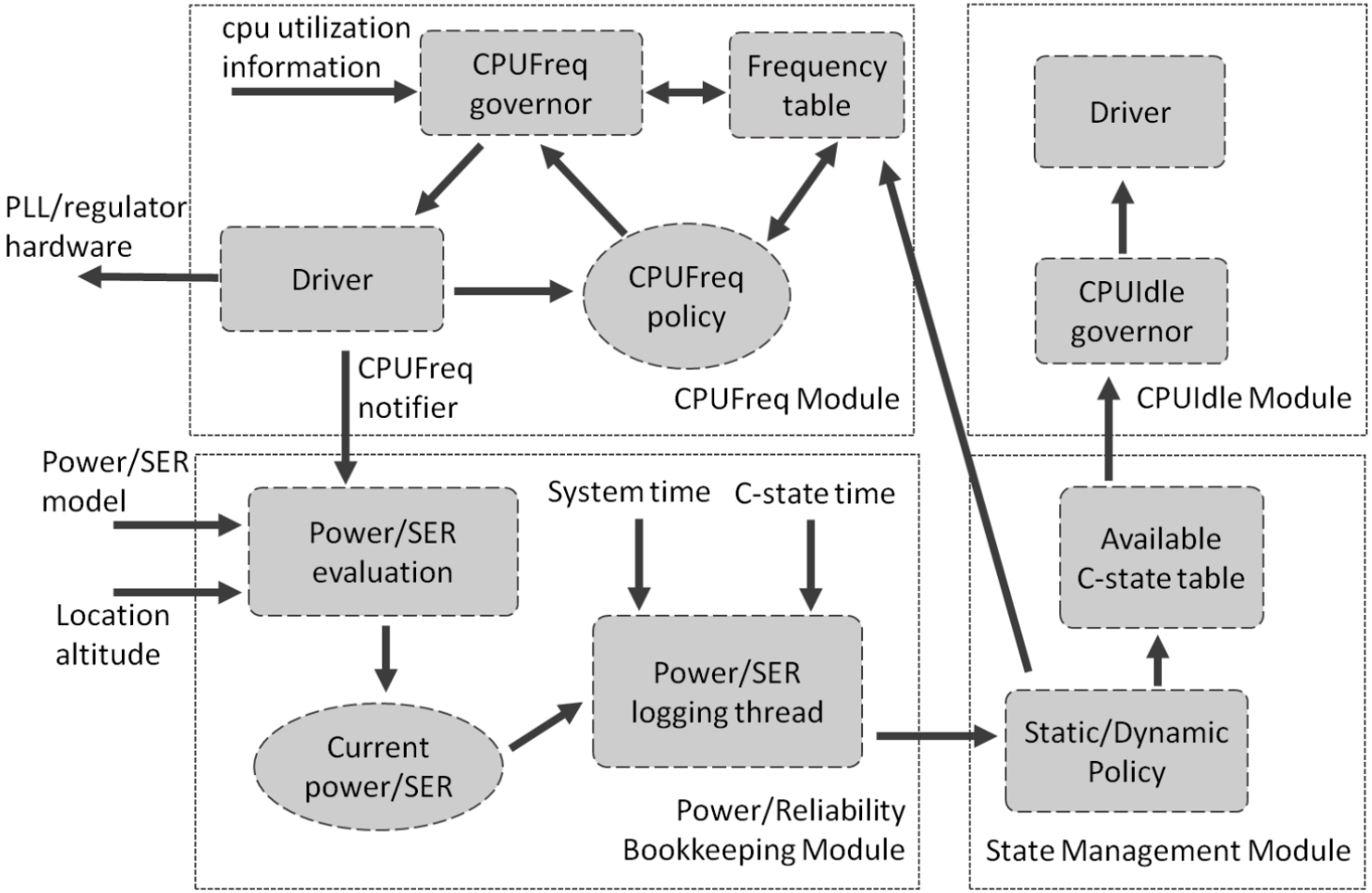
CPUFreq states (P-states)

Frequency	Voltage	Latency
1000 MHz	0.90V	2.1ms
800 MHz	0.82V	2.1ms
720 MHz	0.80V	2.2ms
600 MHz	0.76V	2.2ms
300 MHz	0.72V	3.2ms

CPUIdle states (C-states)

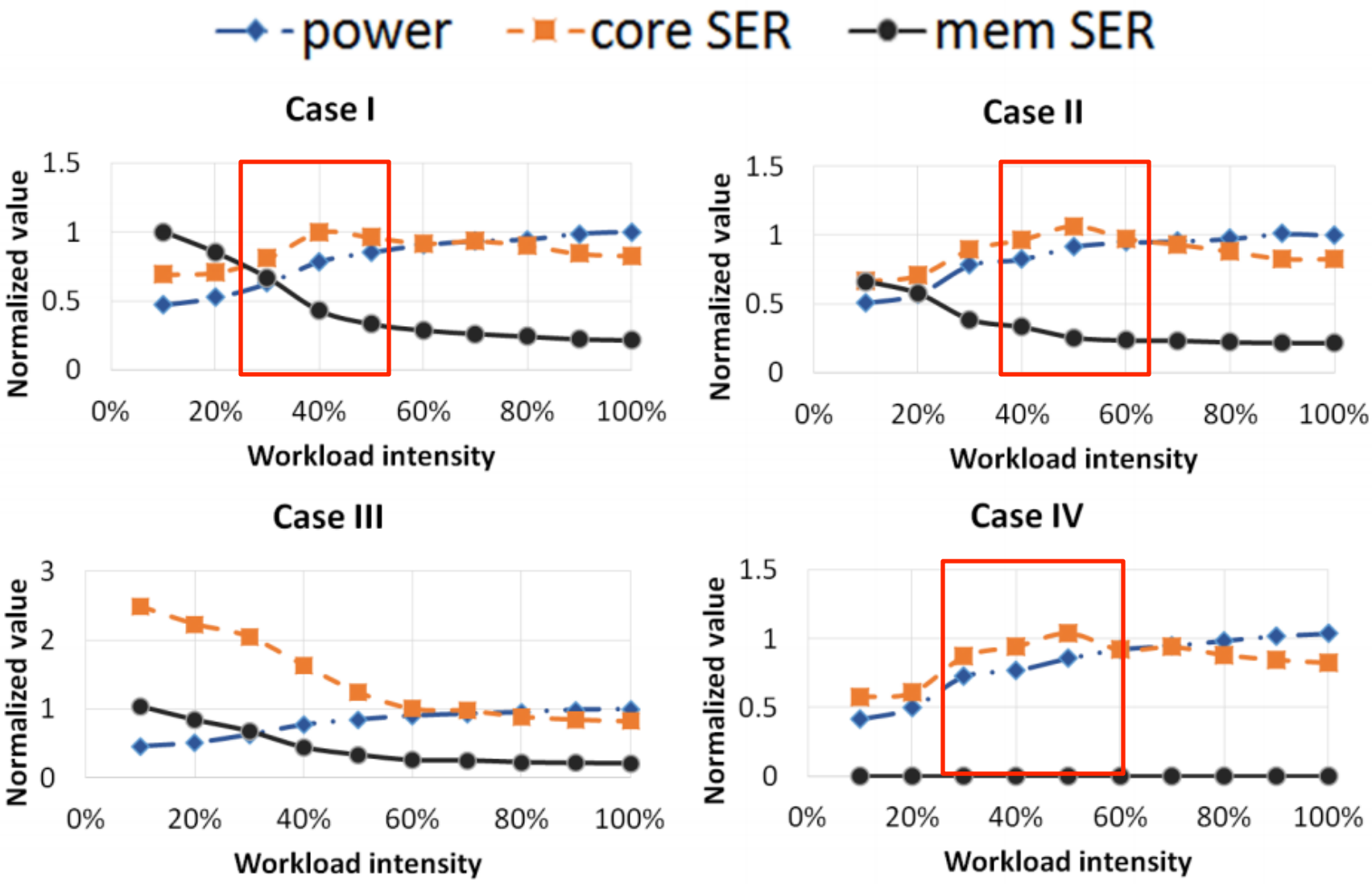
State	Description	Exit Latency	Target Residency
C1	WFI	68us	150us
C2	clock gating	130us	200us
C3	core power gating	530us/1060us[2]	800us/ 1450us
C4	core power gating with SRAM retention	650us/1180us[2]	1000us/ 1550us

Embedding Bookkeeping Module



Available at: https://github.com/nanocad-lab/JIT-RFTS

Reliability Evaluation



Outline

- Motivation
- Evaluating DPM on System Reliability
  - Evaluation Platform
  - Results
- Exploiting DPM for System Reliability
  - Virtual SER monitor
  - Forbidden state based policy
  - Results
- Conclusion

Monitoring SER

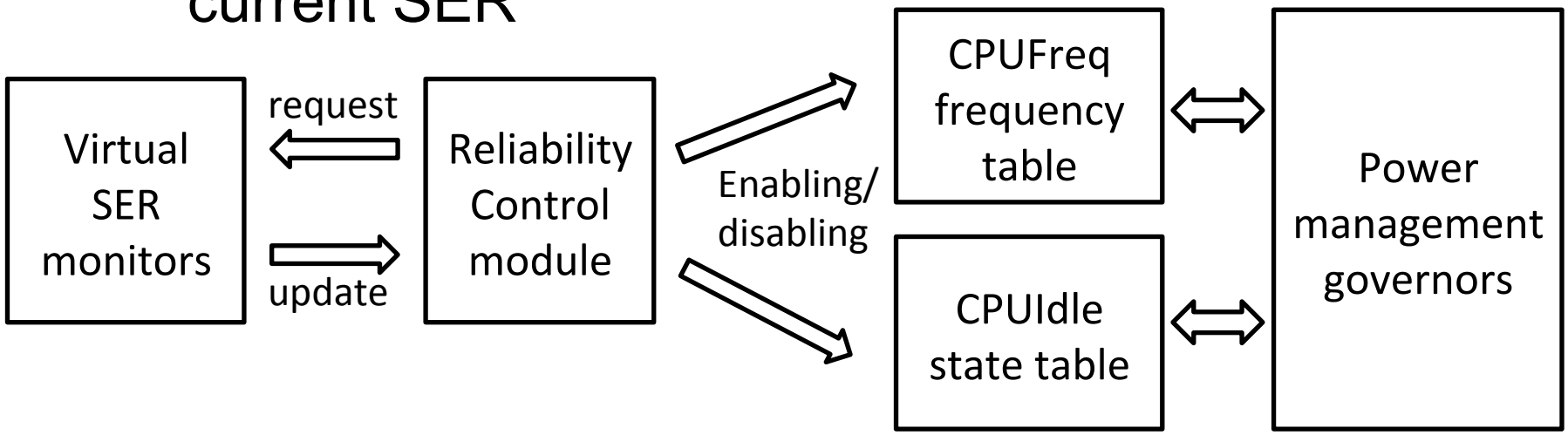
- Soft error rate (SER) depends on the altitude

$$F_{alt}(d) = exp(\frac{d_{SL} - d}{L_n})$$

- Sources of altitude difference
  - Same device used at different locations
  - Same types of device used at different locations
- Hardware SER monitor is infeasible
  - Rare occurrence (< 1 per month per MB)
- System-level Virtual SER monitor
  - Altitude/Location sensors (GPS, barometer etc.)
  - Cloud-based services (network capabilities)

Embedding Reliability Into DPM

- Considerations:
  - Effective
  - Non-intrusive
- Forbidden-state based policy
  - Preserve existing structures
  - Enabling/Disabling power states depending on current SER

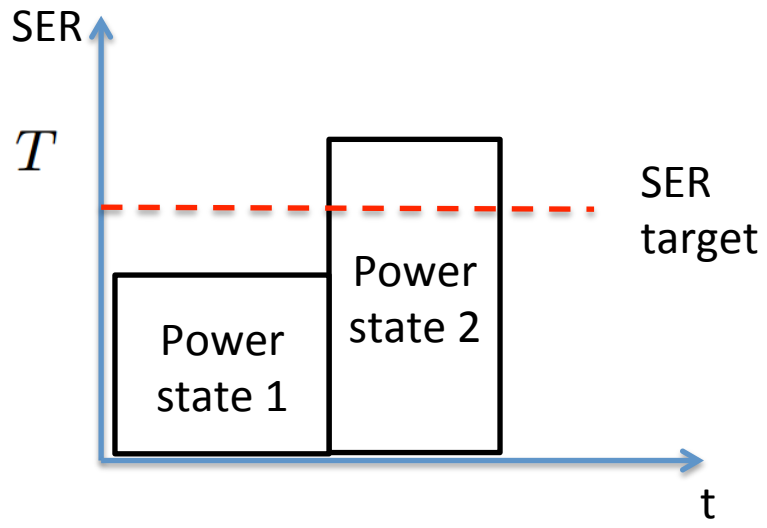


Static/Dynamic Policy

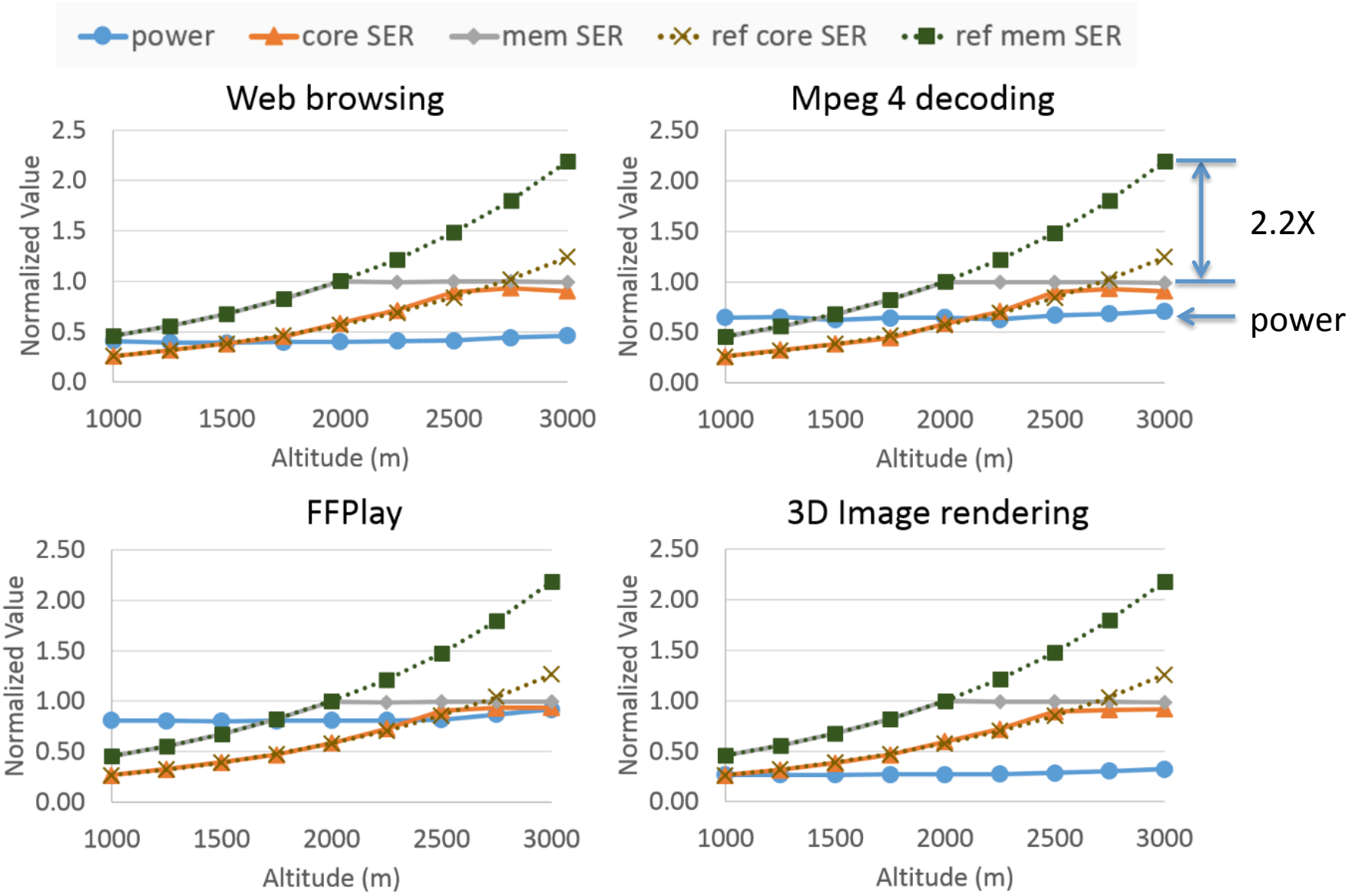
- Forbidden-state based policy
  - Enable/Disable state based on current SER
  - Overly pessimistic to limit instantaneous SER
- Dynamic policy
  - Limit accumulated SER over a fixed period
  - Dynamically enable/disable power states

$$SER_{end} = SER_{acc}(t_{start}) + SER_{avg} \cdot T$$

$$SER_{budget} = \frac{SER_{end} - SER_{acc}(t)}{t_{end} - t}$$



Experimental Results



Policy Effectiveness Evaluation

- Calculating optimal P-state and C-state selection
  - Offline
  - Formulated and solved as ILP
- Compare the power difference between our policy and optimal P-state and C-state selections

Power overhead compared to optimal state selection

Benchmark	P-state optimal		C-state optimal	
	Mean	Max	Mean	Max
Web browsing	0.08%	0.42%	0.72%	2.18%
MPEG 4 decoding	0.14%	0.73%	0.63%	2.47%
FFPlay	0.18%	1.31%	0.58%	2.51%
3D Image rendering	0.14%	1.17%	0.53%	1.89%

Conclusion

- Evaluating DPM on System Reliability
  - Building a evaluation Platform
- Exploiting DPM for System Reliability
  - JIT-RFTS trade-offs
    - 2.2X SER reduction with minimal overhead
  - Simple yet effective policies
    - <3% overhead compared to offline optimal
- Future work will integrate more reliability phenomena
- The kernel module code is available for download at https://github.com/nanocad-lab/JIT-RFTS

Q&A