Adaptive Resource Management through Self-Awareness[+]

Nikil D. Dutt

Center for Embedded and Cyber-Physical Systems (CECS)

University of California, Irvine

dutt@uci.edu

http://www.ics.uci.edu/~dutt

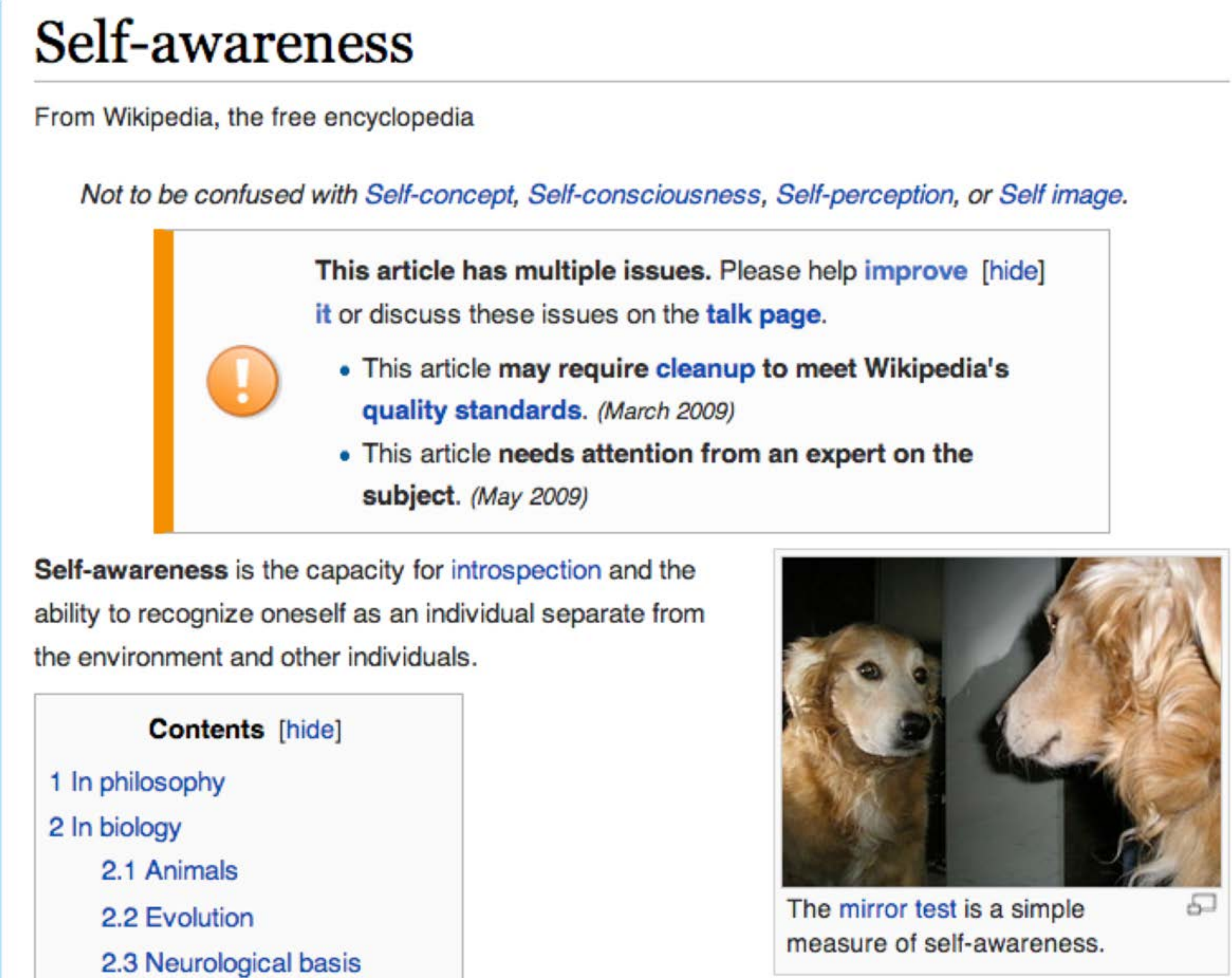
https://duttgroup.ics.uci.edu

[+] Joint work with Tiago Mück, Bryan Donyanavard, Kasra Moazzemi, Amir Rahmani, Santanu Sarma, Biswadip Maity

Dutt Research Group

Research Partially Supported by the National Science Foundation

Self-Awareness ?



Self-awareness

From Wikipedia, the free encyclopedia

Not to be confused with Self-concept, Self-consciousness, Self-perception, or Self image.

This article has multiple issues. Please help improve it or discuss these issues on the talk page. [hide]

- This article **may require cleanup to meet Wikipedia's quality standards**. *(March 2009)*
- This article **needs attention from an expert on the subject**. *(May 2009)*

Self-awareness is the capacity for introspection and the ability to recognize oneself as an individual separate from the environment and other individuals.

The mirror test is a simple measure of self-awareness.

Contents [hide]

1 In philosophy
2 In biology
2.1 Animals
2.2 Evolution
2.3 Neurological basis

Computational Self-* Properties

- **Self-Awareness** [Hinchey2006]: System is aware of its ***self states and behaviors***

- **Context-Awareness** [Parashar 2005] : System is aware of ***context – i.e., its operational environment***

- ***Self-configuring ->*** capability of reconfiguring automatically
- ***Self-healing*** [Robertson2005] ***->*** *self-diagnosing and self-repairing*
- ***Self-optimizing->*** *capability of self-tuning* or *Self-adjusting*
- ***Self-protecting ->*** capability of detecting dangerous outcomes (e.g. security breaches) and recovering from their effects

Outline

- Computational Self-Awareness
- Why Self-Aware Chips?
- Cross-Layer Sensing & Actuation
- Towards Self-Aware Chips
- Supervisory Control & Coordination

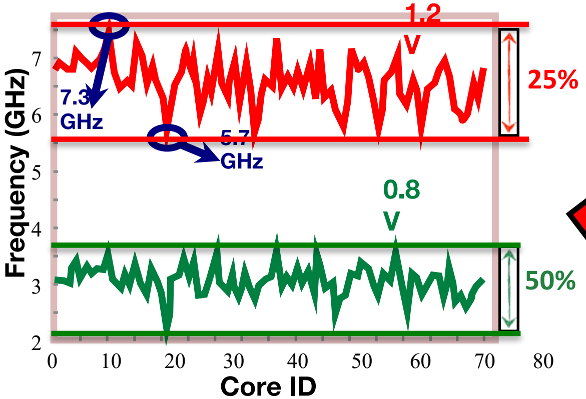
Why On-Chip Self-Awareness (1)?

[Source: NSF Variability Expedition Project]

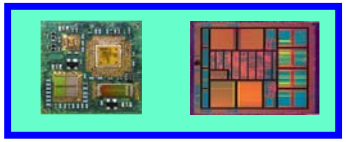
Why On-Chip Self-Awareness (1)?

Variability-induced challenges

Manufacturing



Frequency variation across 80-core processor within a single die in Intel's 65nm node at 1.2V and 0.8V [Dighe10]

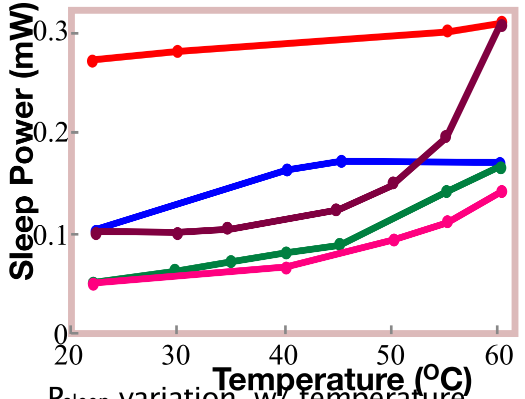


[Source: NSF Variability Expedition Project]

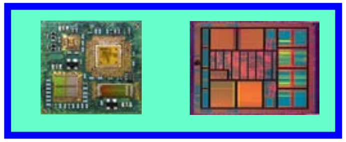
Why On-Chip Self-Awareness (1)?

Variability-induced challenges

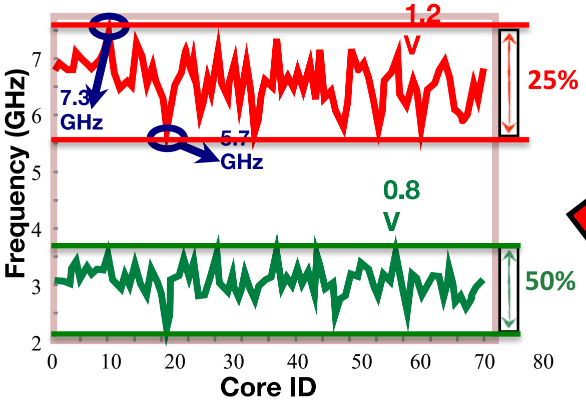
Environment



P_{sleep} variation w/ temperature across 5 ARM Cortex M3 processor instances



Manufacturing



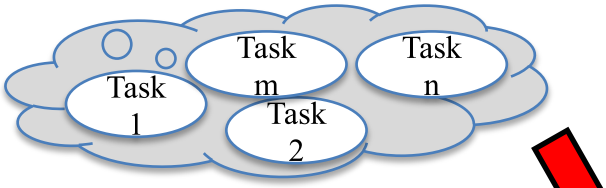
Frequency variation across 80-core processor within a single die in Intel's 65nm node at 1.2V and 0.8V [Dighe10]

[Source: NSF Variability Expedition Project]

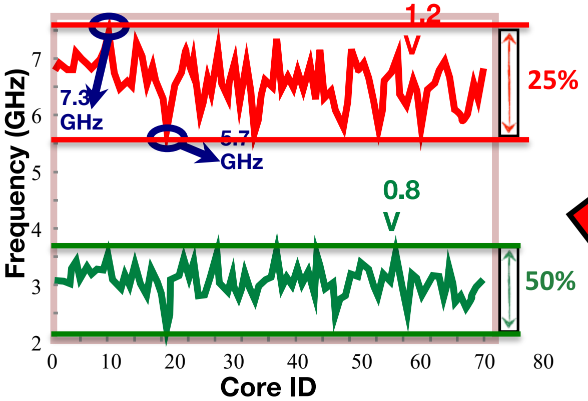
Why On-Chip Self-Awareness (1)?

Variability-induced challenges

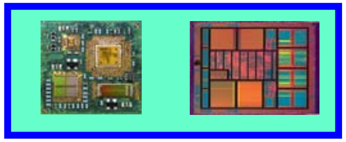
Applications: *varying compute, memory, communication*



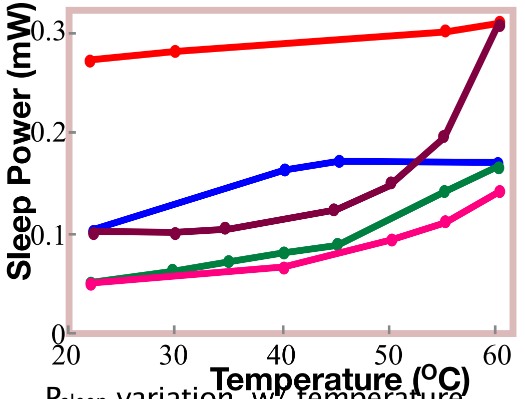
Manufacturing



Frequency variation across 80-core processor within a single die in Intel's 65nm node at 1.2V and 0.8V [Dighe10]



Environment



P_{sleep} variation w/ temperature across 5 ARM Cortex M3 processor instances

Triple Whammy!

[Source: NSF Variability Expedition Project]

Why On-Chip Self-Awareness (2) ?

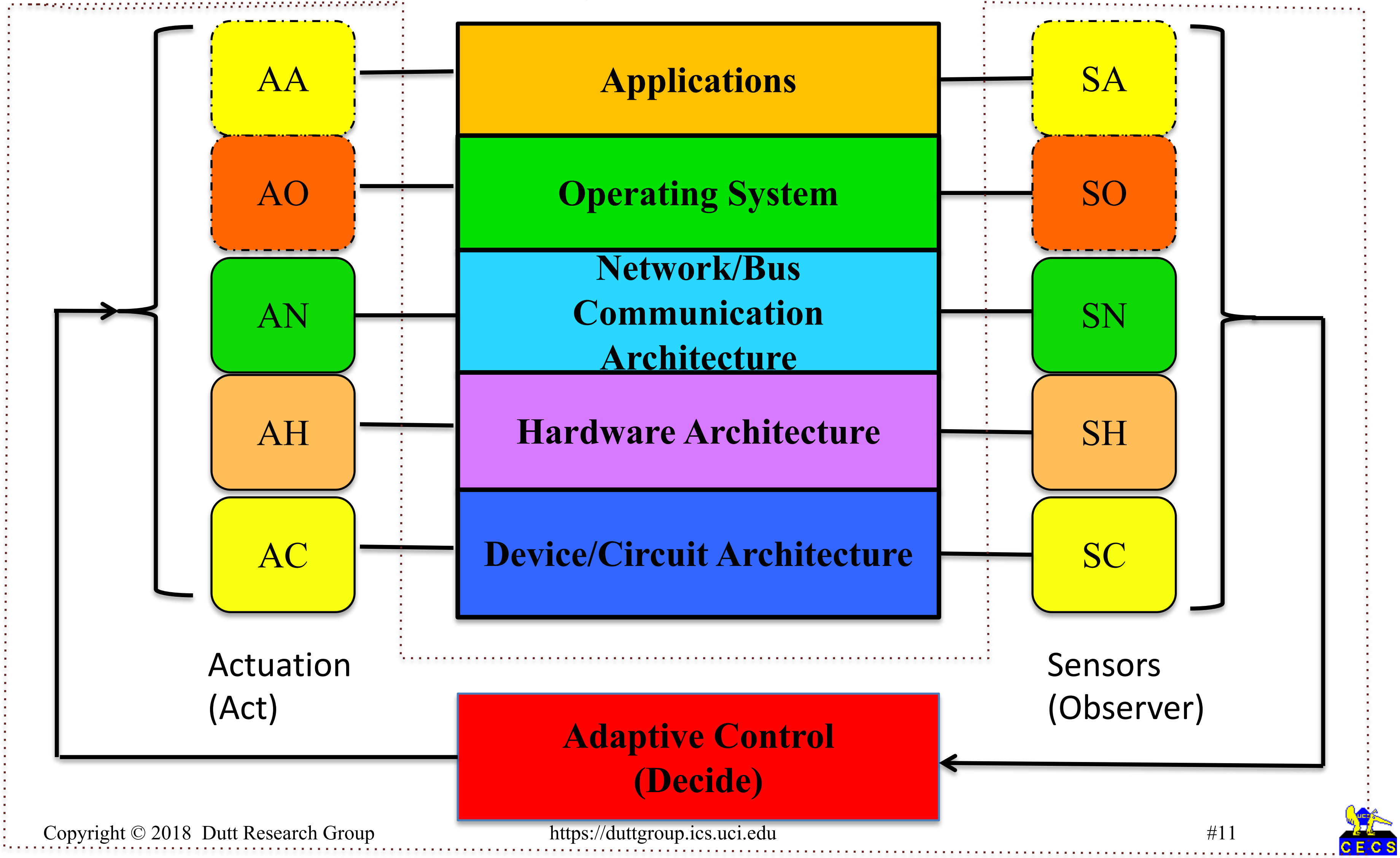
- Chips must adapt to:
  - Performance, Power, Resilience, Security,….
- Provide Guarantees
- Dynamically manage multi-dimensional trade-offs
  - Performance, Power/Energy, Thermal,…..
  - QoS, TDP, Wear-out, ….

Exploit Computational Self-Awareness

Outline

- Computational Self-Awareness
- Why Self-Aware Chips?
- Cross-Layer Sensing & Actuation
- Towards Self-Aware Chips
- Supervisory Control & Coordination

Cross-Layer Physical/Virtual Sensing & Actuation



Examples of Virtual Sensors and Actuators Across Layers of CPSoC

Layers	Virtual/Physical Sensors	Virtual/Physical Actuators
Application	Execution Time, Workload Power, Energy,	Loop perforation Algorithmic Choice
Operating System	System Utilization Peripheral States	Task Allocation, Scheduling, Migration, Duty Cycling
Network/Bus Communication	Bandwidth; Packet/Flit status; Channel Status, Congestion, Latency	Adaptive Routing Dynamic Bandwidth Allocation Ch. no and direction
Hardware Architecture	Cache misses, Miss rate; access rate; IPC, Throughput, ILP/MLP, Core asymmetry	Cache Sizing; Reconfiguration, Resource Provision Static/Dynamic Redundancy
Circuit/Device	Circuit Delay, Aging, leakage Temperature, oxide breakdown	DVFS, DFS, DVS ABB, Clock and Power-gating

Outline

- Computational Self-Awareness
- Why Self-Aware Chips?
- Cross-Layer Sensing & Actuation
- Towards Self-Aware Chips
- Supervisory Control & Coordination

Self-Reflection & Introspection

- Ability to create a ***self-model (introspect)***
- Ability to model their own body/structure (usually known ***self-modeling***)
- Ability to model their own ***behavior***
- ***Metacognition capacity***: *'models one's own thinking', 'think about thinking'*
- *System with* ***two/multiple minds:*** *one* ***being modeled*** *and other* ***doing modeling***

Reflex vs Reflect

Reflexive, Reactive

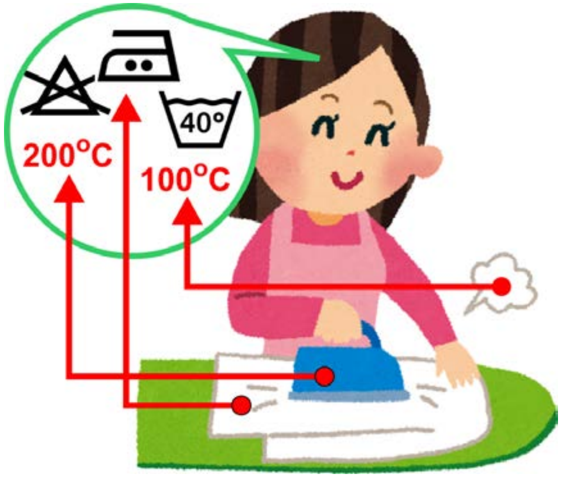
- **Actions driven solely on external feedback**
  - **E.g., our autonomic nervous systems**

Reflex vs Reflect

Reflexive, Reactive

- Actions driven solely on external feedback
  - E.g., our autonomic nervous systems

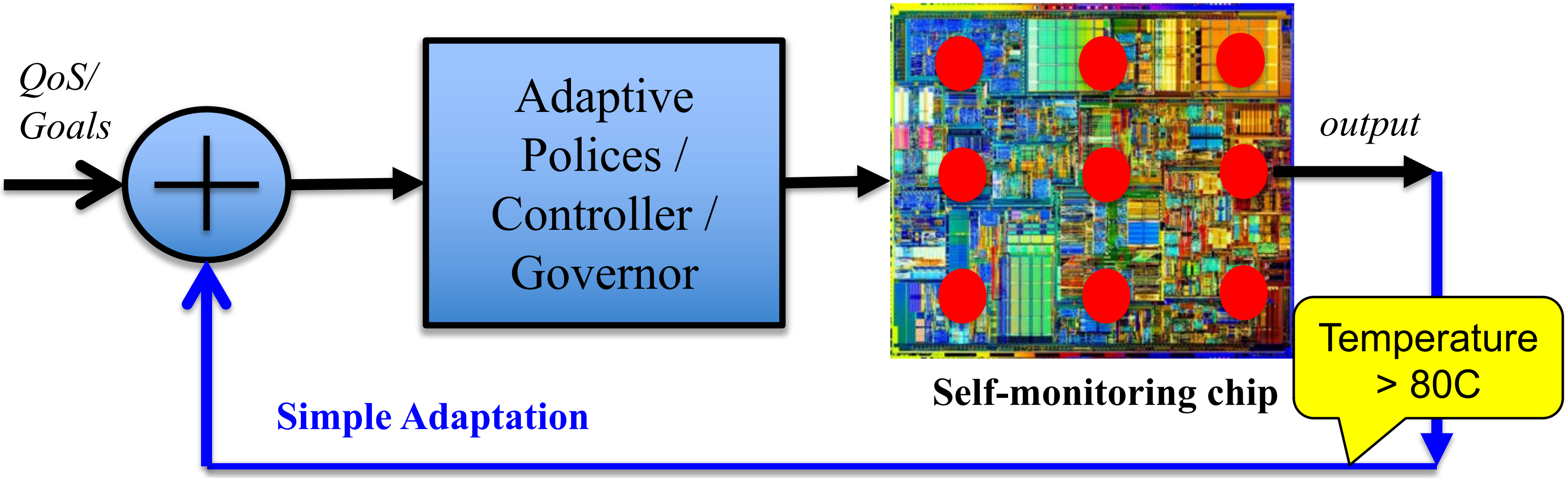
Reflection, Introspection



- Consider past and future outcomes
  - E.g., planning, strategies, policies, …

Towards Self-Aware Chips: What we do now

Reflexive, Reactive



QoS/ Goals

Adaptive Polices / Controller / Governor

output

Temperature > 80C

Self-monitoring chip

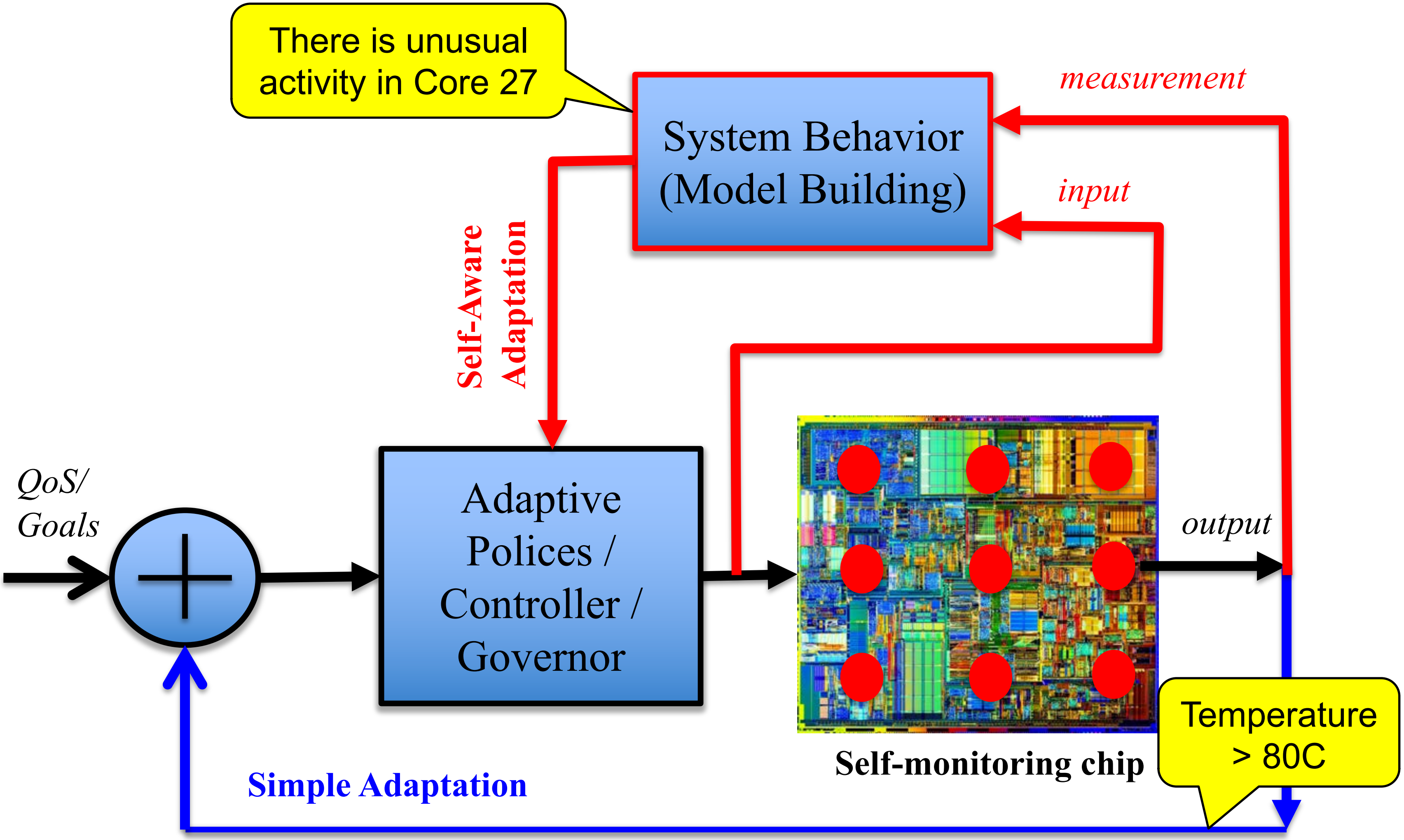
Simple Adaptation

Self-monitoring

[Sarma14, CODES+ISSS14]

Towards Self-aware chips

Beyond simple reactive models

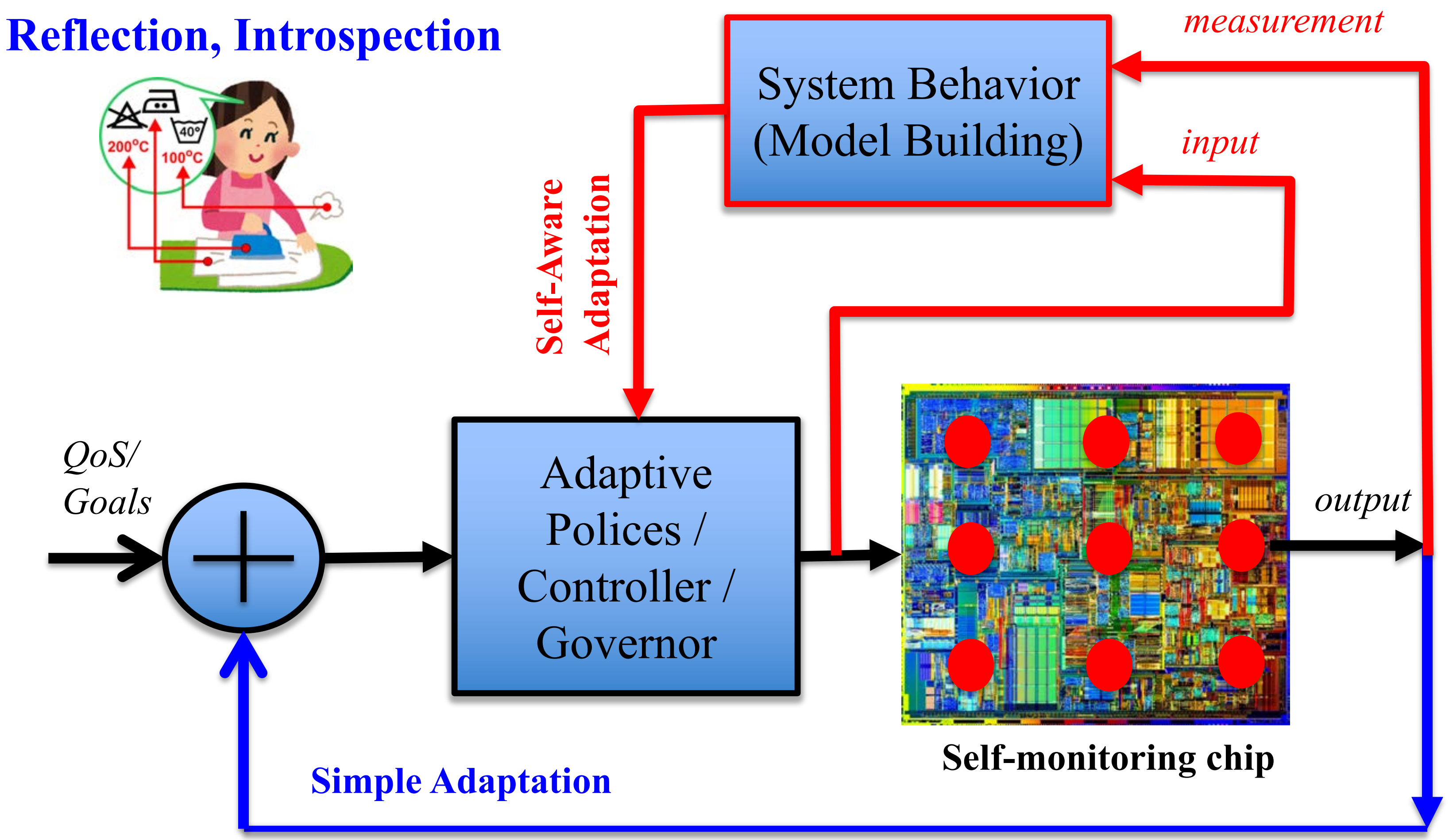

There is unusual activity in Core 27
System Behavior (Model Building)
measurement
input
Self-Aware Adaptation
QoS/ Goals
Adaptive Polices / Controller / Governor
output
Self-monitoring chip
Temperature > 80C
Simple Adaptation

Self-monitoring and **Self-modeling**

[Sarma14, CODES+ISSS14]

Towards Self-aware chips

Beyond simple reactive models

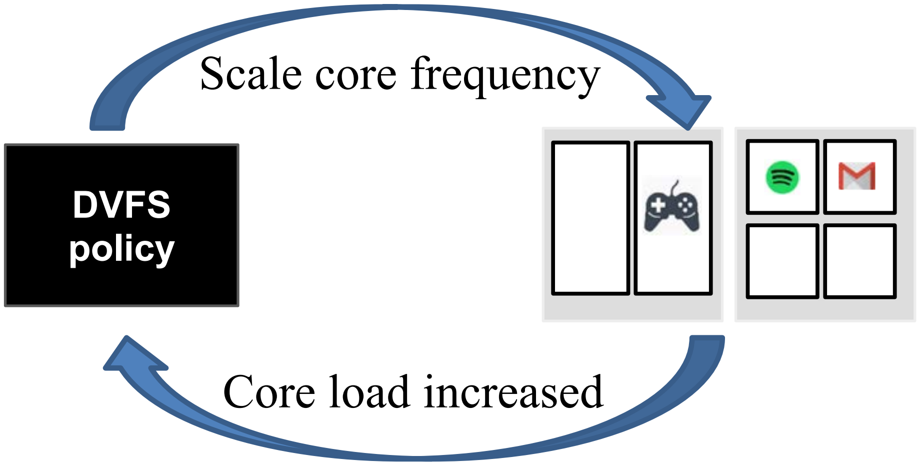


Self-monitoring and **Self-modeling**

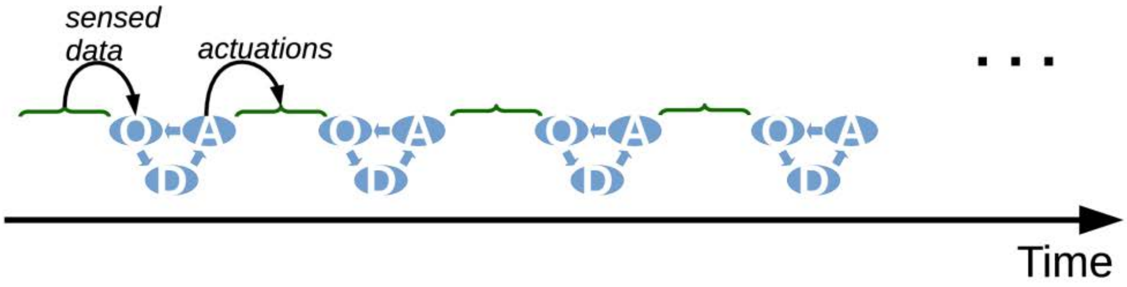
[Sarma14, CODES+ISSS14]

Today: “Reflexive” Resource Management

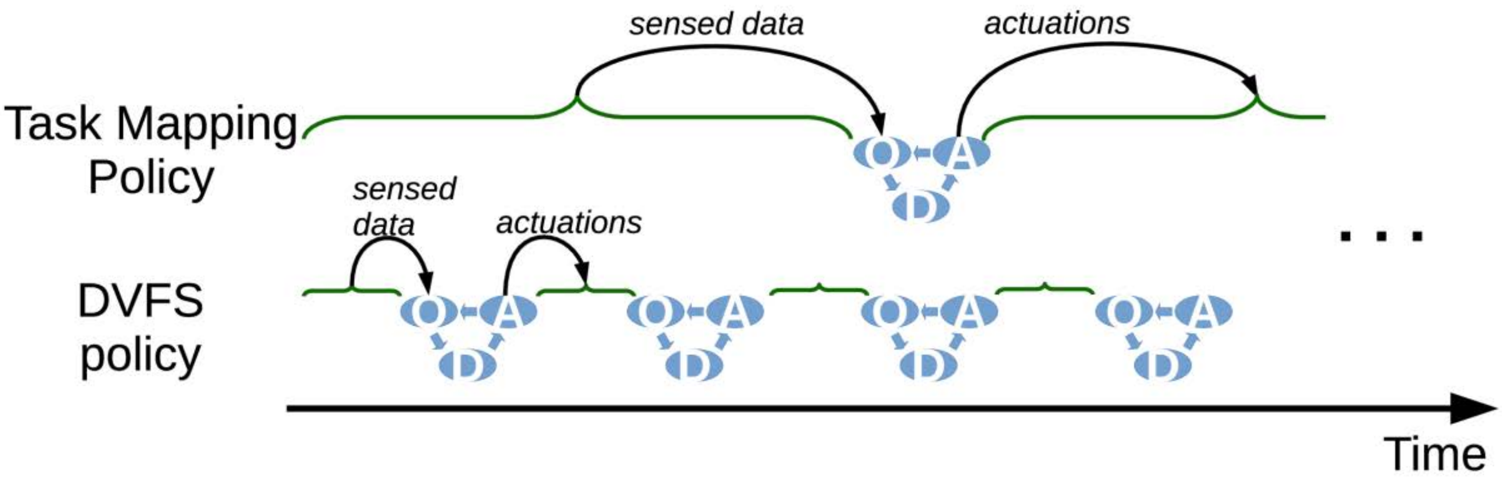
- Dynamic Voltage/Frequency scaling (DVFS)



- Observe-Decide-Adapt approaches

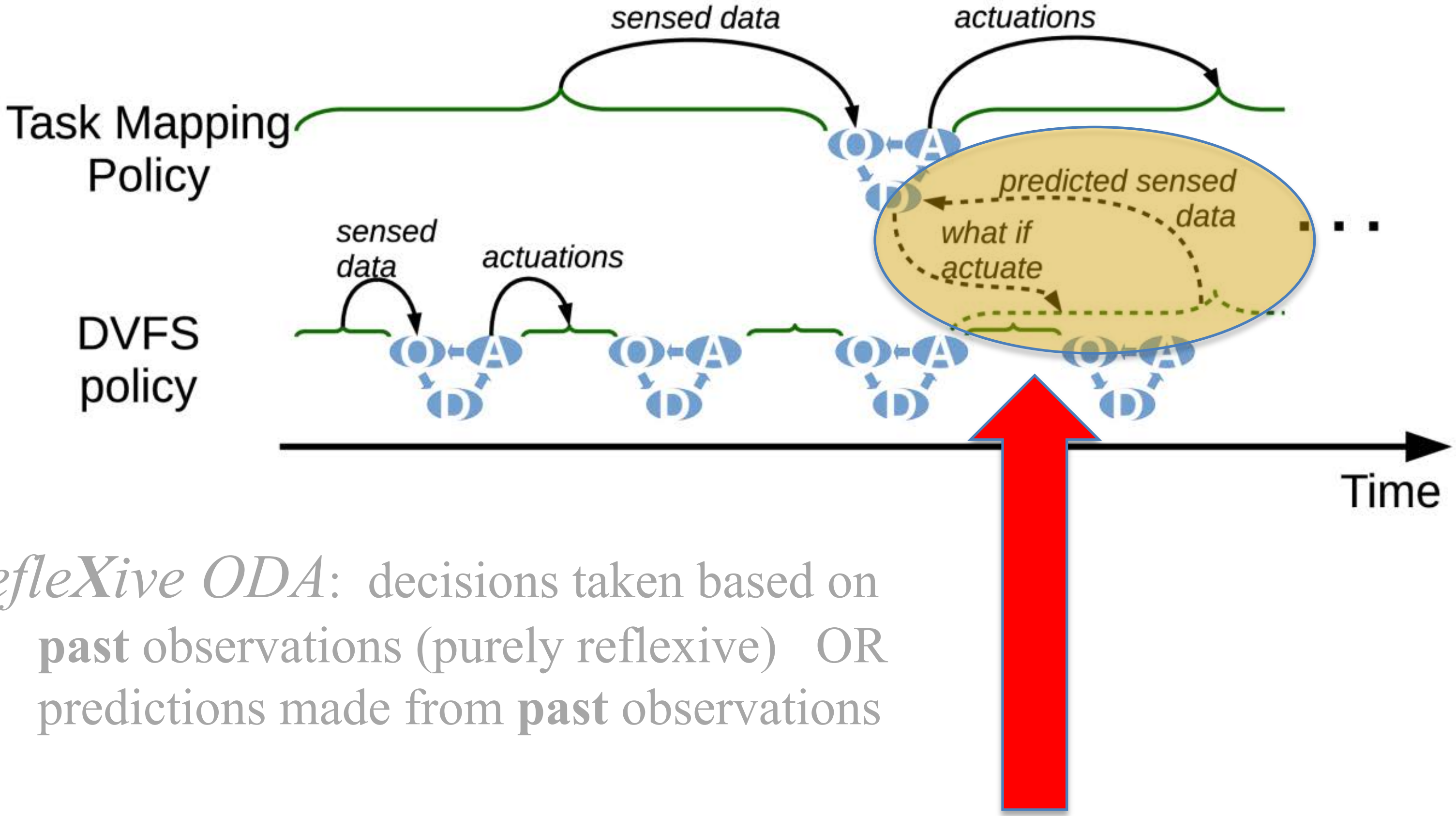


RefleXive vs RefleCTive Resource Management



- *RefleXive ODA*: decisions taken based on
  - **past** observations (purely reflexive) OR
  - predictions made from **past** observations

RefleXive vs RefleCTive Resource Management



- *RefleXive ODA*: decisions taken based on
  - **past** observations (purely reflexive) OR
  - predictions made from **past** observations

- **RefleCTive approach**: considers **future** events that could happen in the next iteration of the ODA loop

Adaptive Resource Management

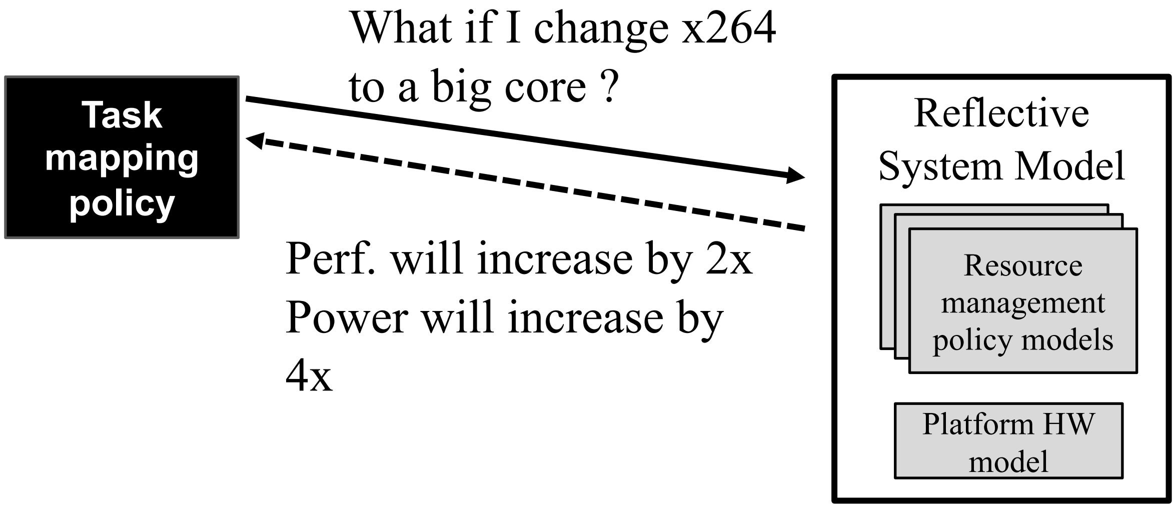
- Use concept of **reflection**
  - **Reflection**: *change your actions based on both external feedback and* ***introspection*** *(i.e., self-assessment)*

Adaptive Resource Management

- Use concept of **reflection**
  - Reflection: *change your actions based on both external feedback and* ***introspection*** *(i.e., self-assessment)*
- **Reflective resource management combines:**
  - Current system state assessed from sensing information (e.g., readings from performance counters, power sensors, etc.)
  - Models to predict the behavior of other system components before performing an action

MARS: Our coordination approach

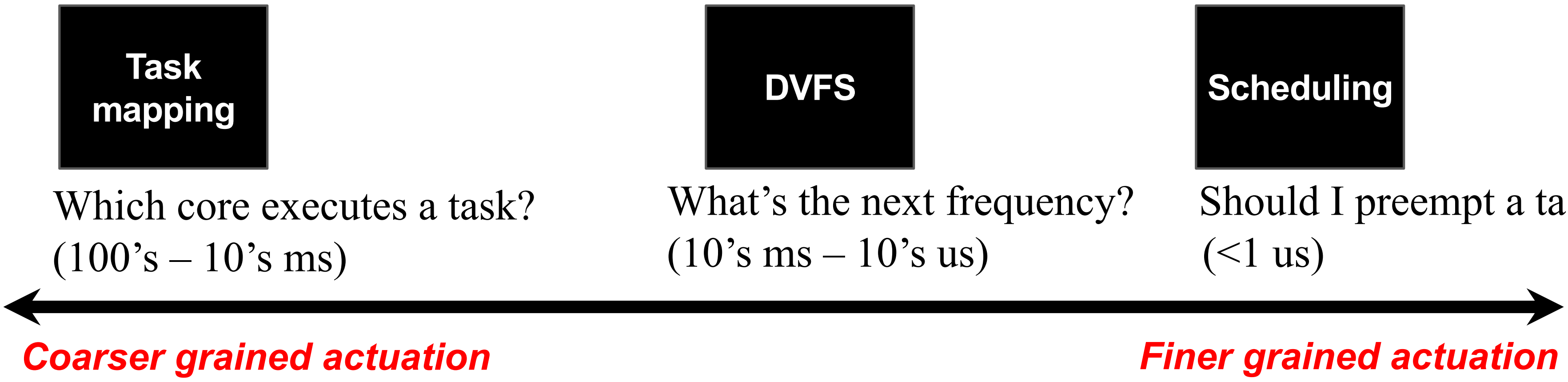
- Coordination though **reflective** resource management
  - **MARS: M**iddleware for **A**daptive **R**eflective **S**ystems



Task mapping policy
What if I change x264 to a big core ?
Perf. will increase by 2x
Power will increase by 4x
Reflective System Model
Resource management policy models
Platform HW model

Do we have room for reflection ?

- Systems actuations happen at different timescales



Task mapping

Which core executes a task?
(100's – 10's ms)

DVFS

What's the next frequency?
(10's ms – 10's us)

Scheduling

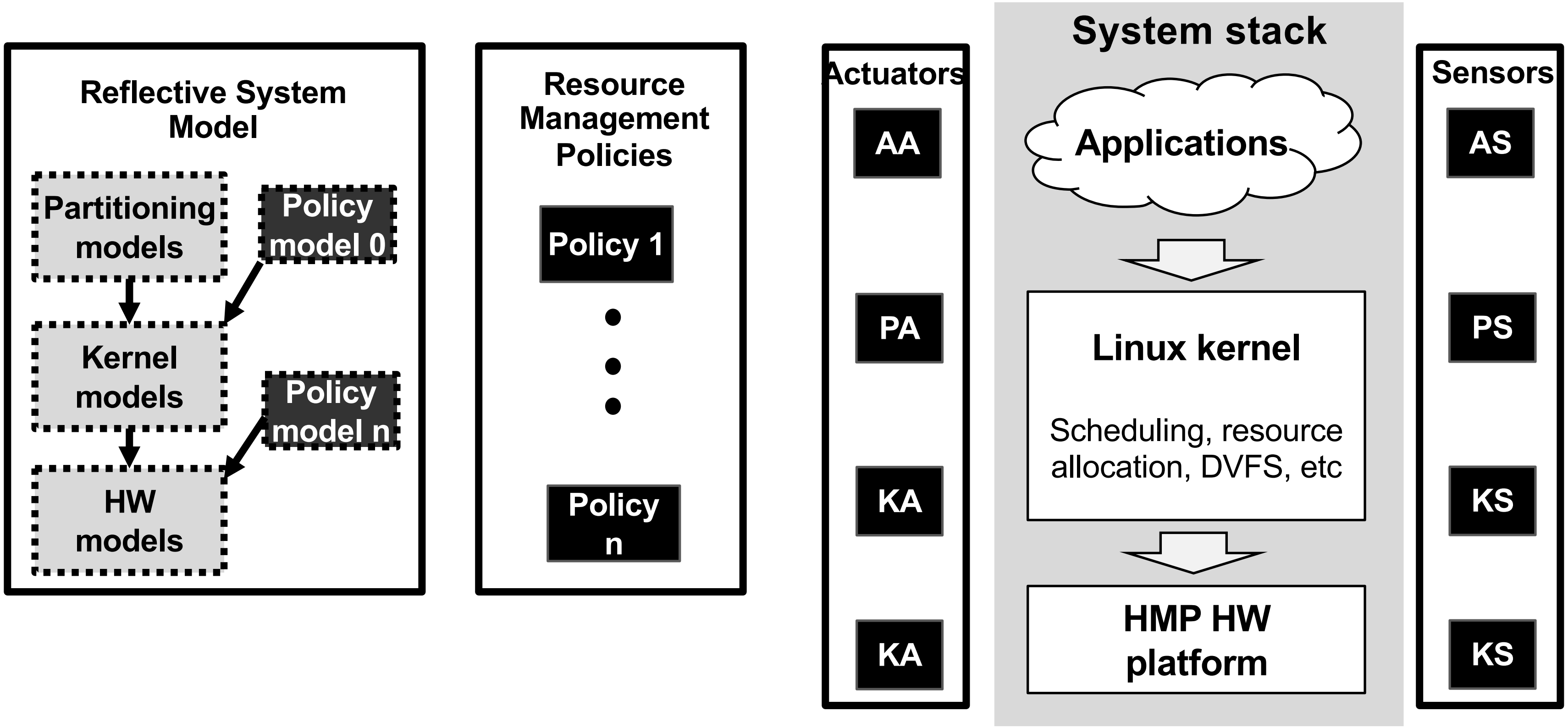
Should I preempt a task ?
(<1 us)

Coarser grained actuation

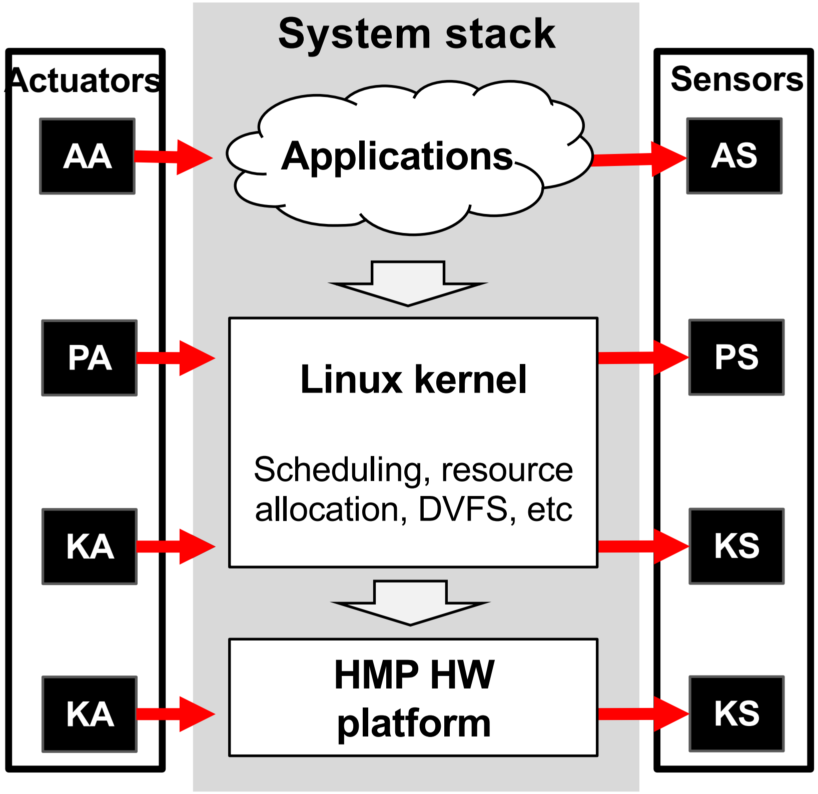
Finer grained actuation

- Some actuations happen quickly with little room for reasoning
- Other actuations can occur on larger timescales
  - **Task mapping, Wear-leveling (for aging)….**

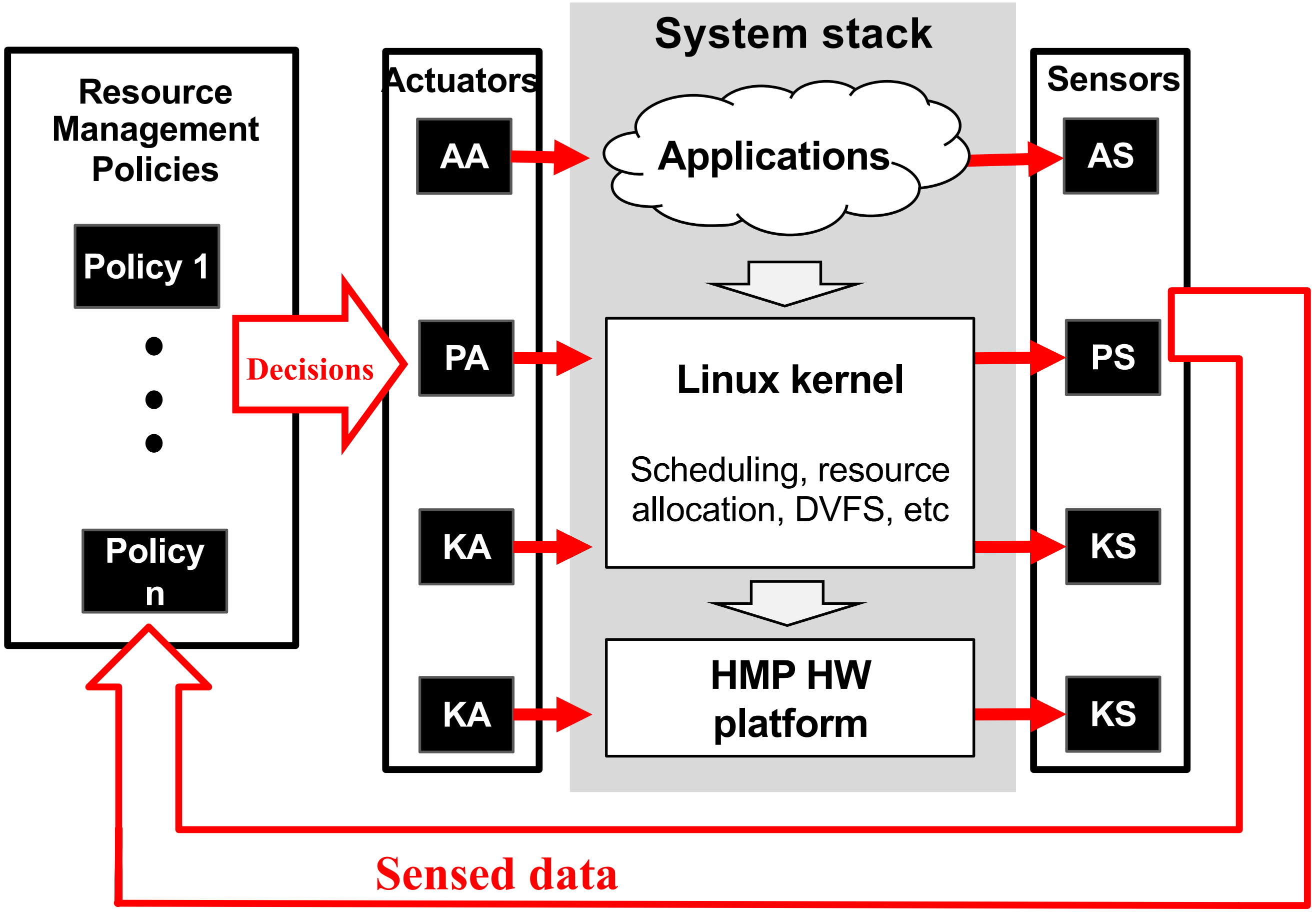
MARS middleware for reflective resource management



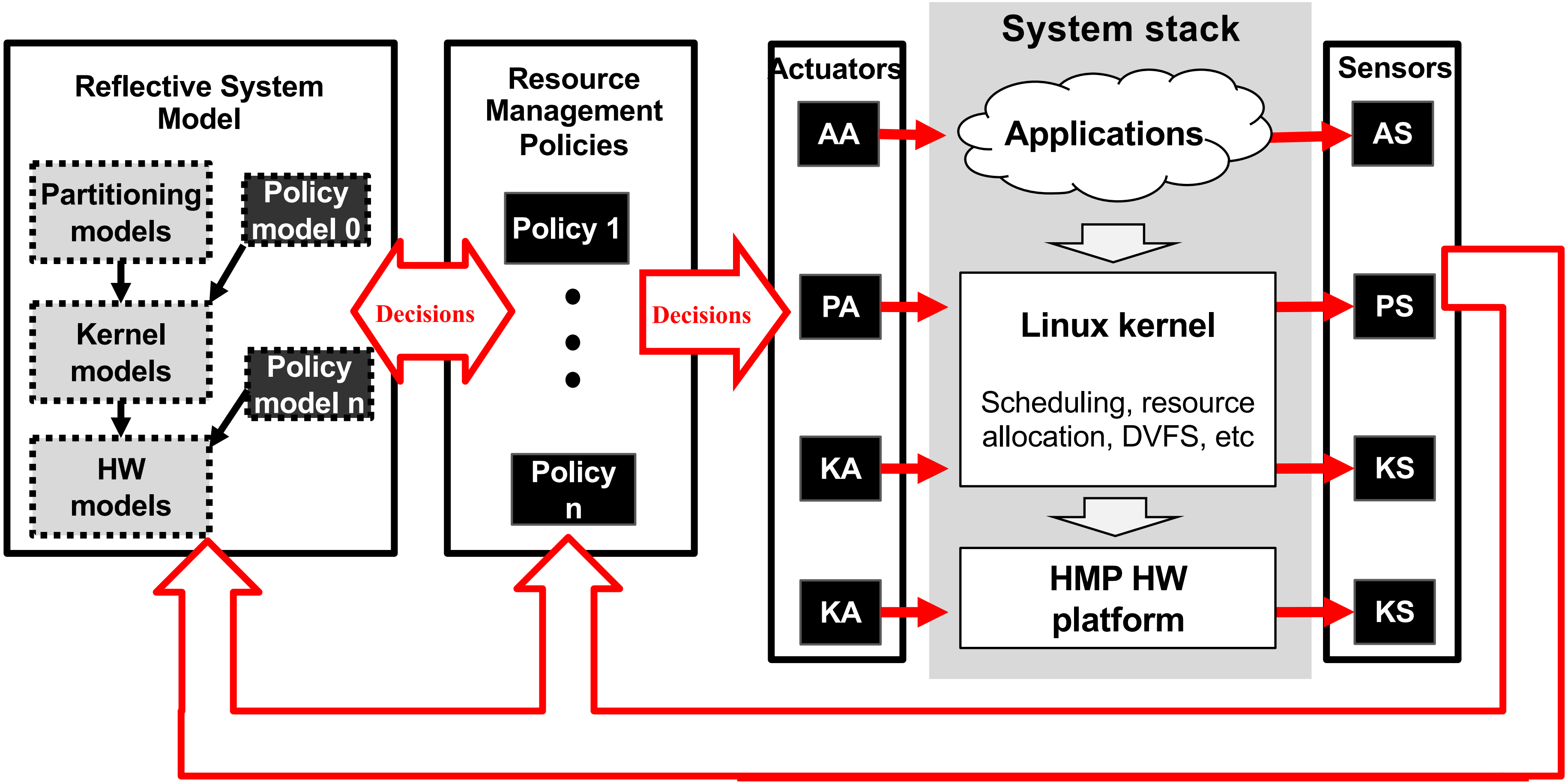
MARS middleware for reflective resource management



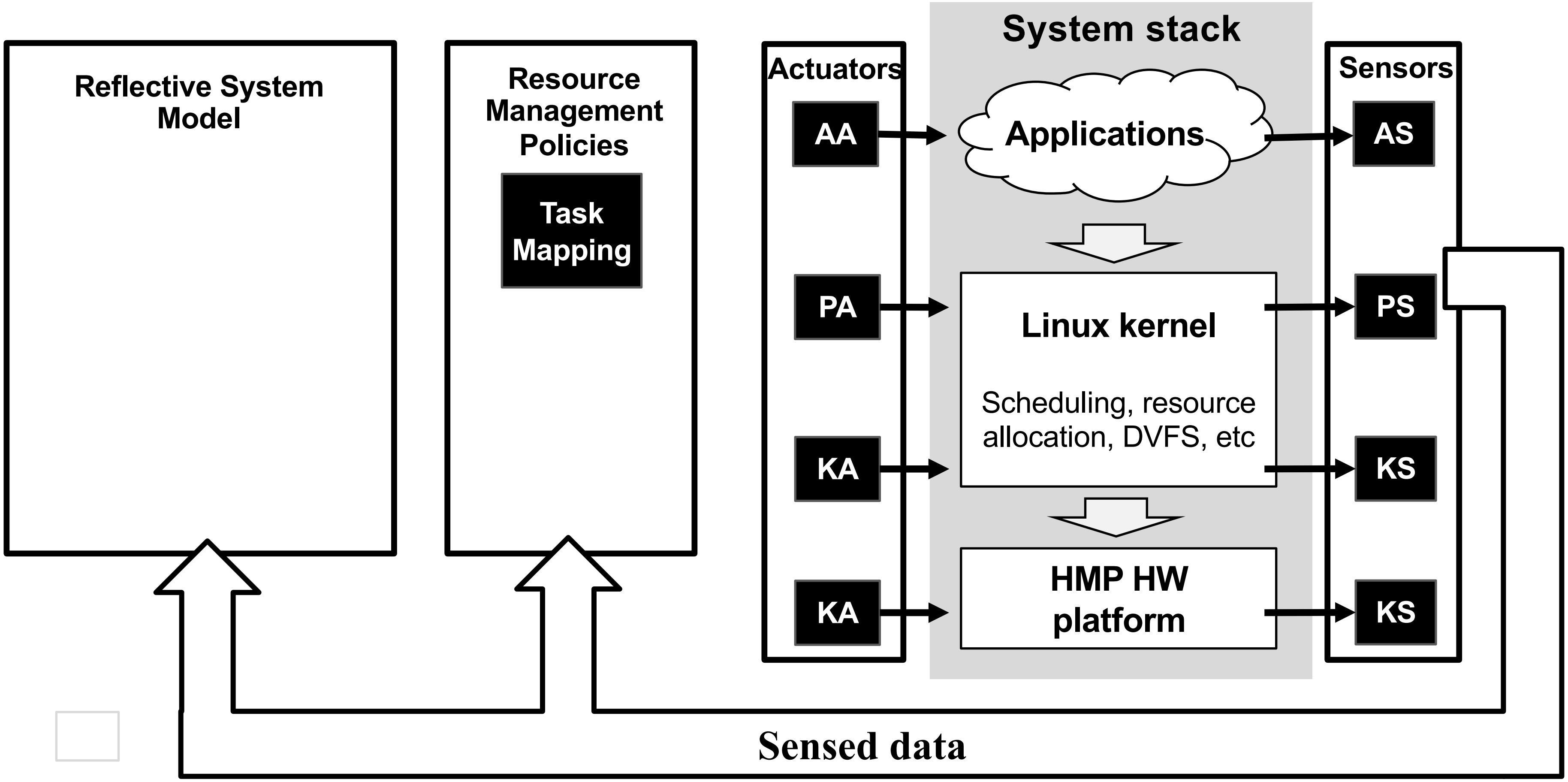
MARS middleware for reflective resource management



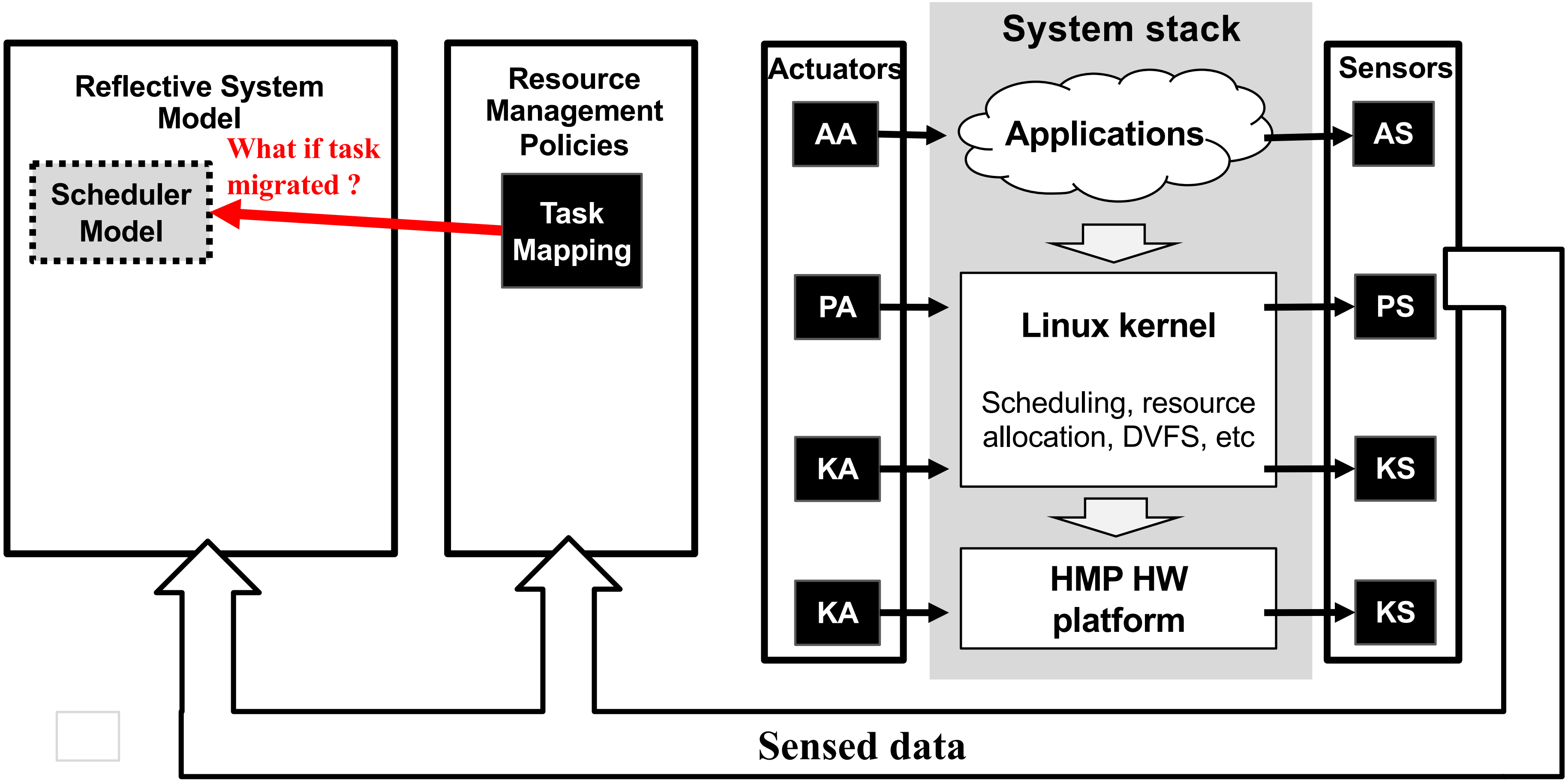
MARS middleware for reflective resource management



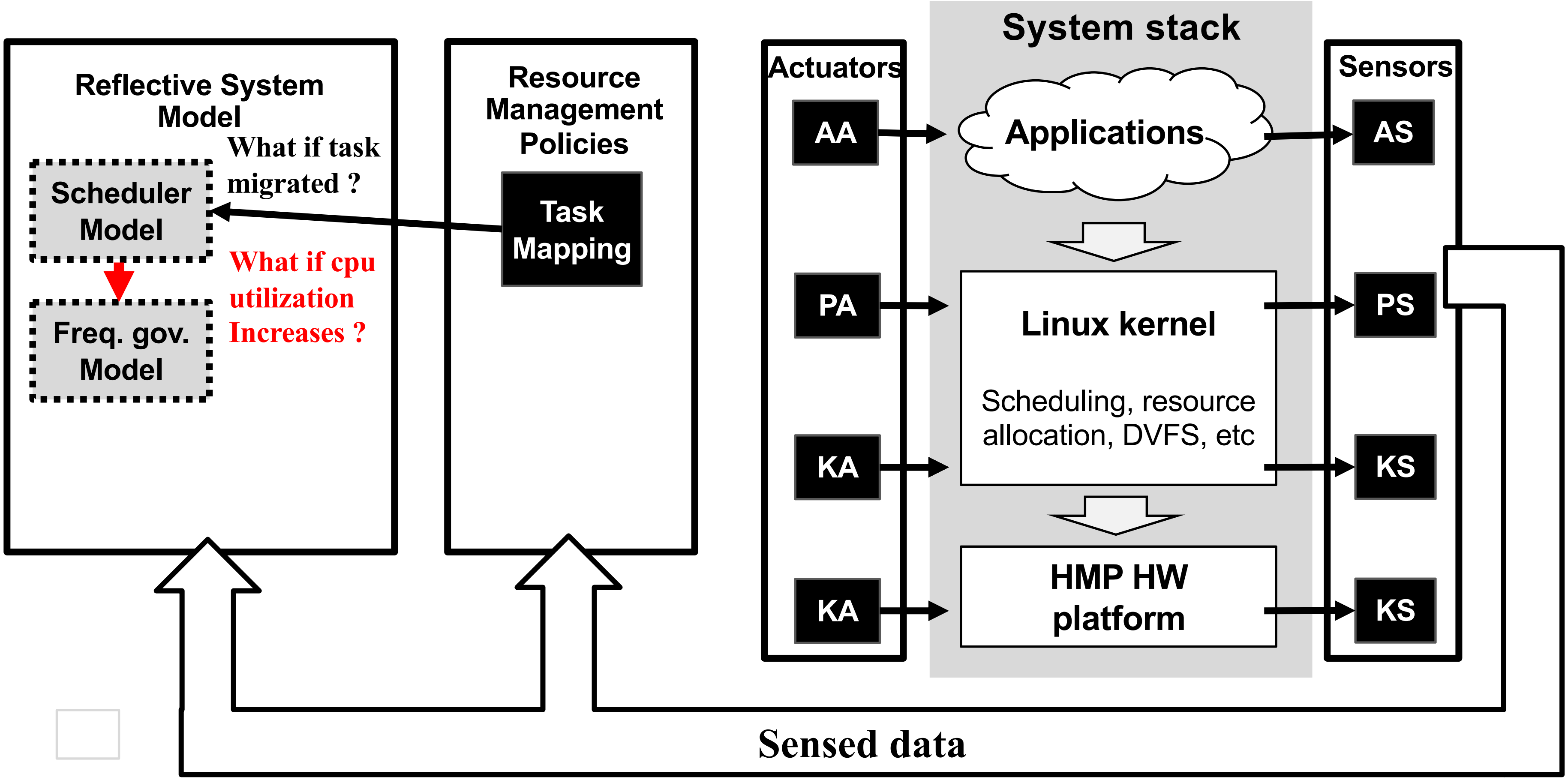
Example: Reflective Task Mapping



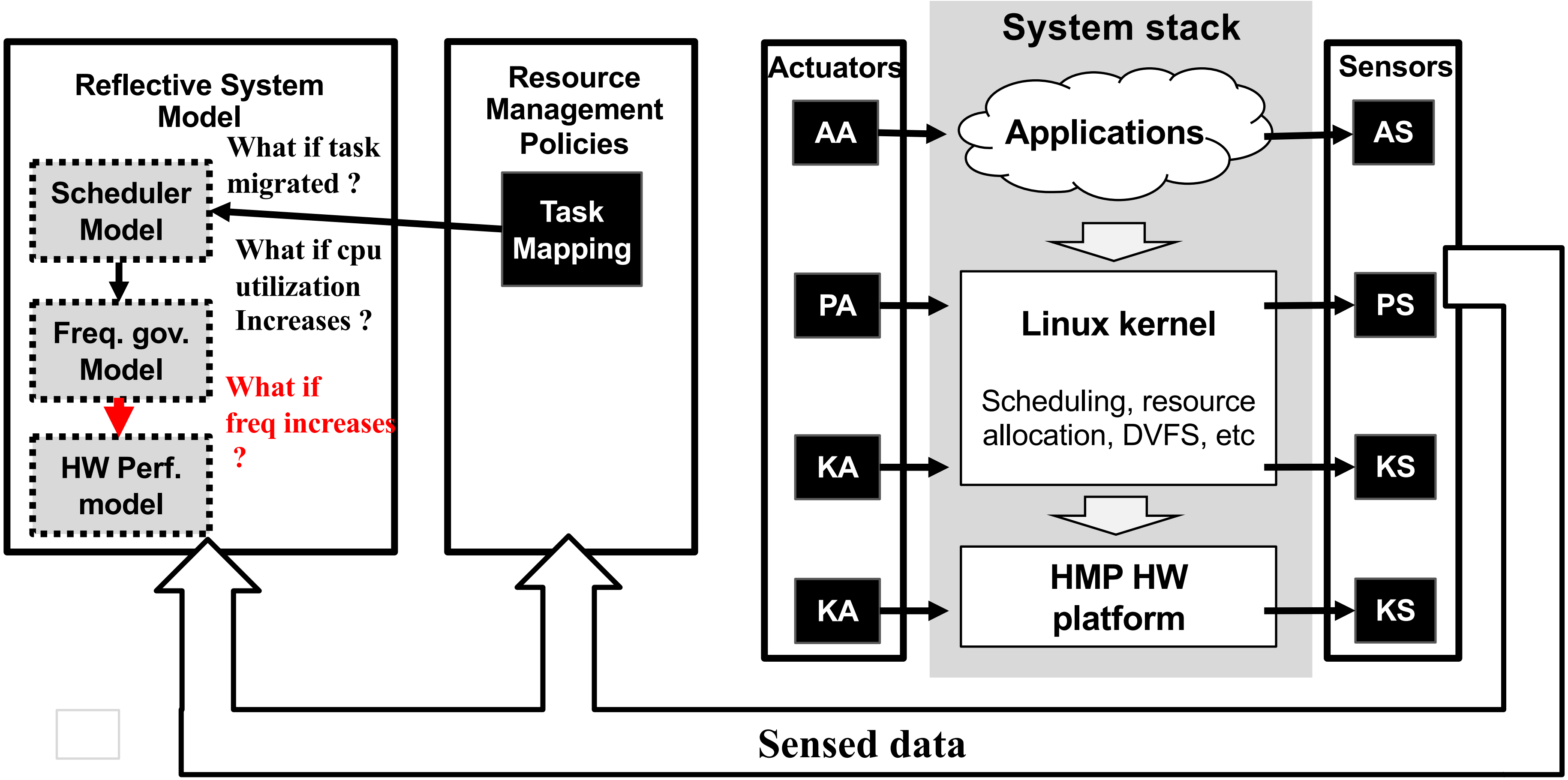
Example: Reflective Task Mapping



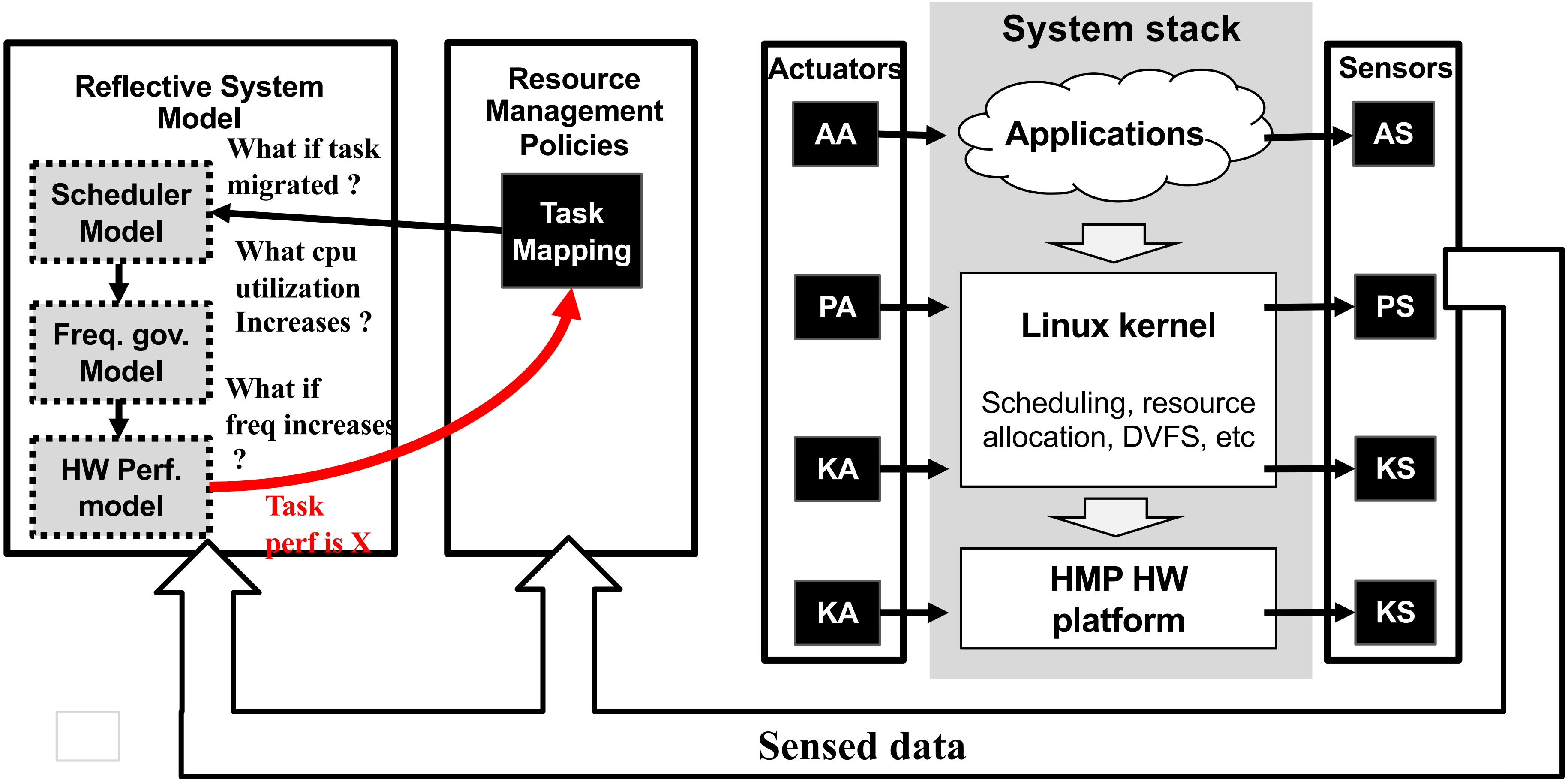
Example: Reflective Task Mapping



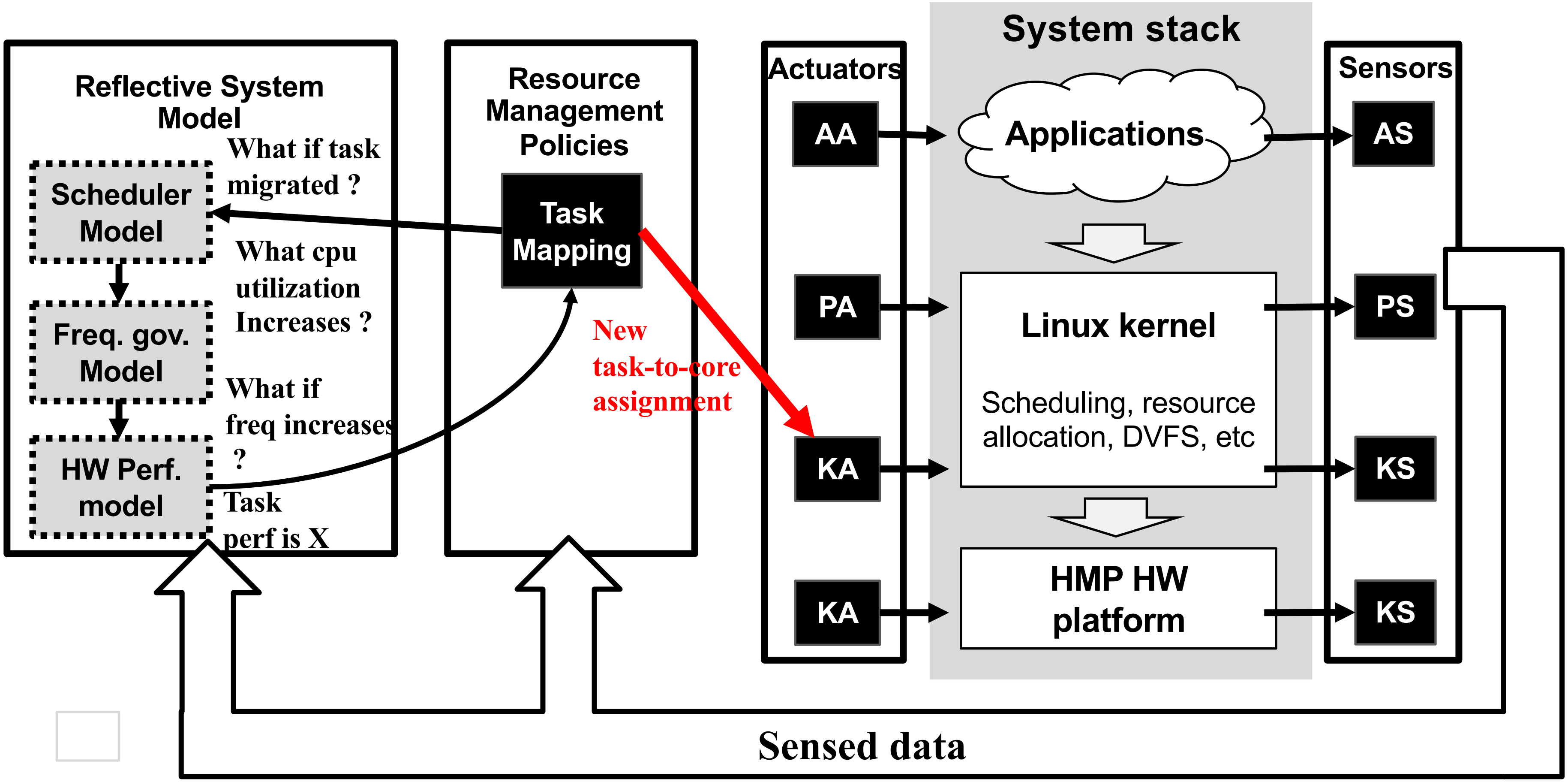
Example: Reflective Task Mapping



Example: Reflective Task Mapping

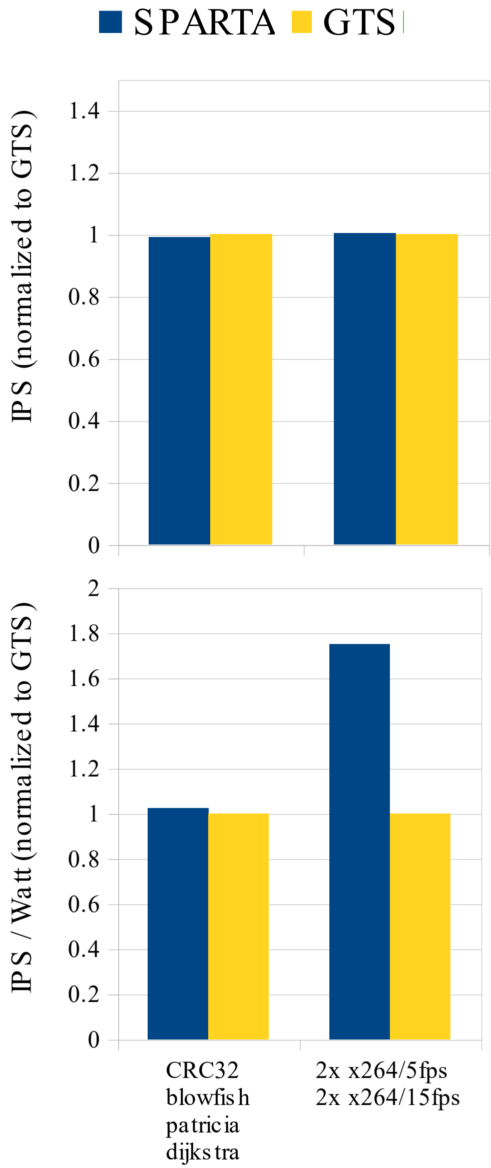


Example: Reflective Task Mapping



SPARTA improvements

- 8-core big.LITTLE Exynos SoC
  - 4x big
  - 4x LITTLE
- Workload mixes (4 tasks each)
  - Mibench
  - x264 (Parsec)
- SPARTA vs Linux's GTS
- **Avg. improvements of 16% in energy efficiency without performance degradation**



Donyanavard, B., Mück, T., Sarma, S., & Dutt, N., *SPARTA: Runtime Task Allocation for Energy Efficient Heterogeneous Many-cores.* CODES+ISSS '16

MARS: Middleware for Adaptive Reflective Computer Systems

- Framework and tools for developing reflective resource/power management policies
  - Use models to predict system behavior
  - Enable easy adaptation to runtime changes
  - Case studies show promise

MARS framework is open source

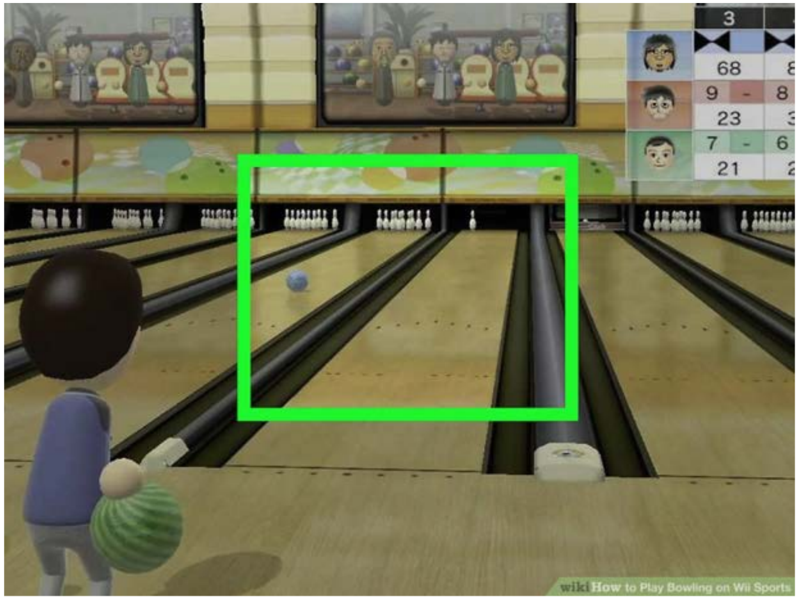
https://github.com/duttresearchgroup/MARS

Outline

- Computational Self-Awareness
- Why Self-Aware Chips?
- Cross-Layer Sensing & Actuation
- Towards Self-Aware Chips
- Supervisory Control & Coordination

Goals and Autonomy

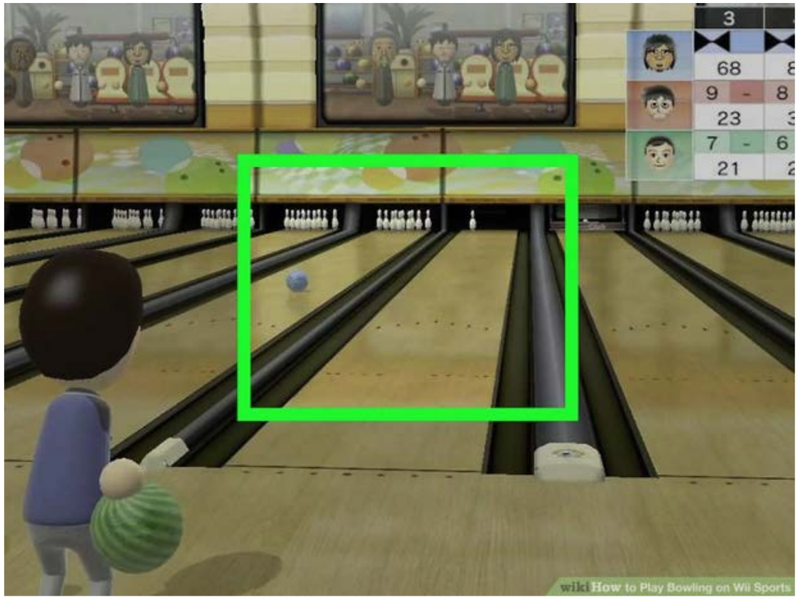
Goal



- **Single, straightforward objective**
  - **E.g., hit the pin**

Goals and Autonomy

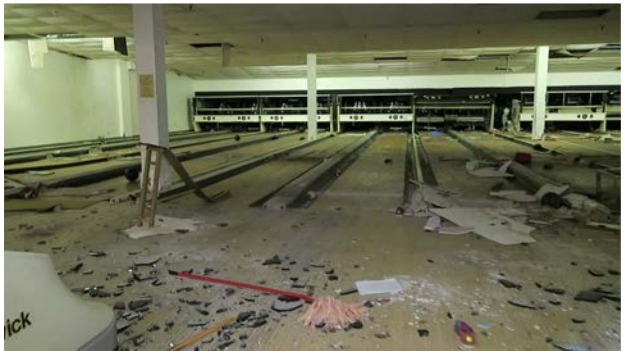
Goal



- Single, straightforward objective
  - E.g., hit the pin

+

Model Imperfection



- What happens when we introduce unpredictability?
  - E.g., balls with different sizes, shapes weights; uneven or damaged surfaces

Goals and Autonomy

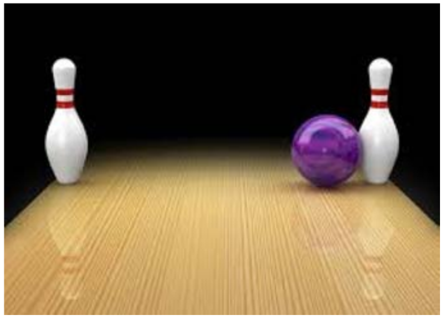
Supervision

- Constrain behavior so we are always headed toward the goal
  - E.g., bumpers

Goals and Autonomy

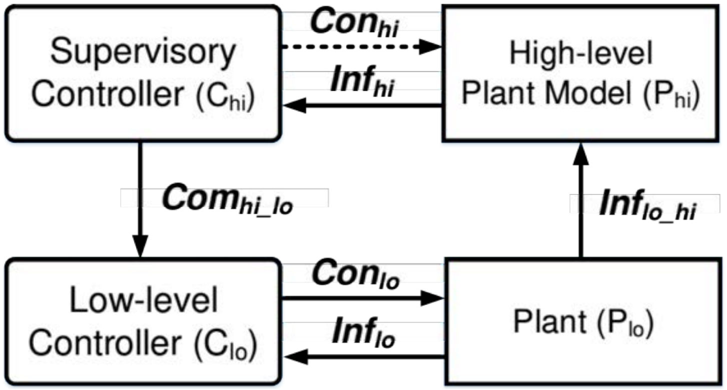
Supervision

- Constrain behavior so we are always headed toward the goal
  - E.g., bumpers
- Bonus: what about when we have more complex or multiple goals?



SPECTR: Our Supervisory Approach

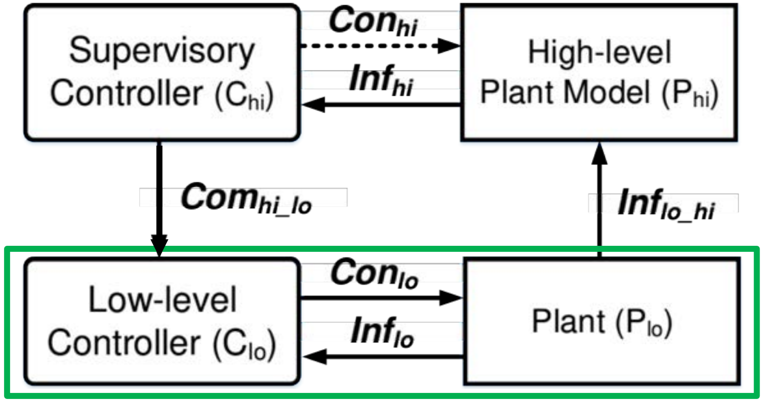
- Autonomy and robustness through supervisory control


Supervisory Controller (C_{hi})
Con_{hi}
Inf_{hi}
High-level Plant Model (P_{hi})
Com_{hi_lo}
Inf_{lo_hi}
Low-level Controller (C_{lo})
Con_{lo}
Inf_{lo}
Plant (P_{lo})

*Rahmani, A. M., Donyanavard, B., Mück, T., Moazzemi, K., Jantsch, A., Mutlu, O., & Dutt, N., *SPECTR: Formal Supervisory Control and Coordination for Many-core Systems Resource Management*. ASPLOS '18

SPECTR: Our Supervisory Approach

- Autonomy and robustness through supervisory control



Supervisory Controller (C_{hi})
Con_{hi}
Inf_{hi}
High-level Plant Model (P_{hi})
Com_{hi_lo}
Inf_{lo_hi}
Low-level Controller (C_{lo})
Con_{lo}
Inf_{lo}
Plant (P_{lo})

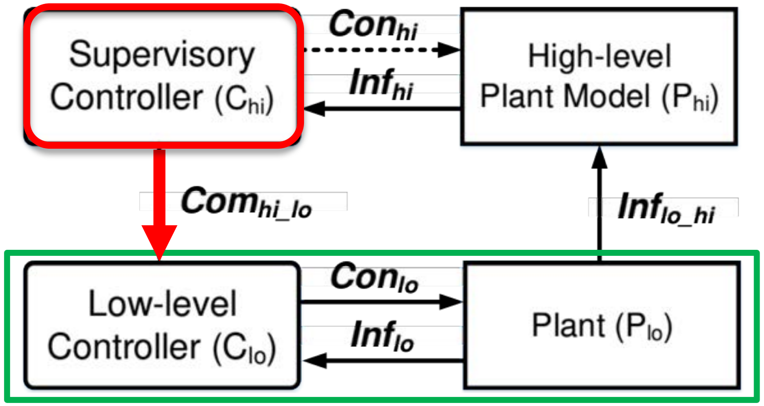
Low-level controllers satisfy objective

*Rahmani, A. M., Donyanavard, B., Mück, T., Moazzemi, K., Jantsch, A., Mutlu, O., & Dutt, N., *SPECTR: Formal Supervisory Control and Coordination for Many-core Systems Resource Management*. ASPLOS '18

SPECTR: Our Supervisory Approach

- Autonomy and robustness through supervisory control

Supervisor bounds behavior of controllers, manages goal


Supervisory Controller (C_{hi})
Con_{hi}
Inf_{hi}
High-level Plant Model (P_{hi})
Com_{hi_lo}
Inf_{lo_hi}
Low-level Controller (C_{lo})
Con_{lo}
Inf_{lo}
Plant (P_{lo})

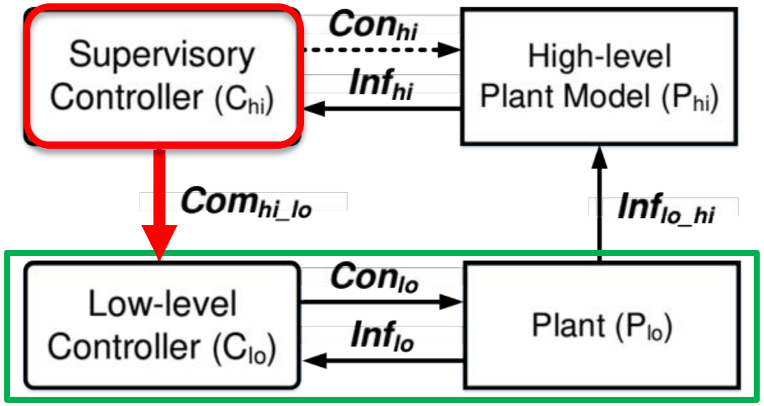
Low-level controllers satisfy objective

*Rahmani, A. M., Donyanavard, B., Mück, T., Moazzemi, K., Jantsch, A., Mutlu, O., & Dutt, N., *SPECTR: Formal Supervisory Control and Coordination for Many-core Systems Resource Management*. ASPLOS '18

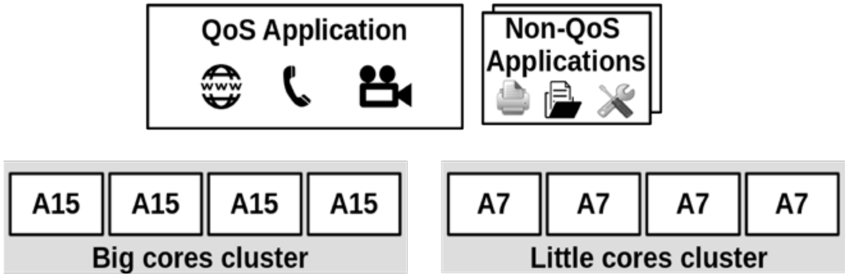
SPECTR: Our Supervisory Approach

- Autonomy and robustness through supervisory control
- Case Study*

Supervisor bounds behavior of controllers, manages goal


Supervisory Controller (C_{hi})
Con_{hi}
Inf_{hi}
High-level Plant Model (P_{hi})
Com_{hi_lo}
Inf_{lo_hi}
Low-level Controller (C_{lo})
Con_{lo}
Inf_{lo}
Plant (P_{lo})

Low-level controllers satisfy objective



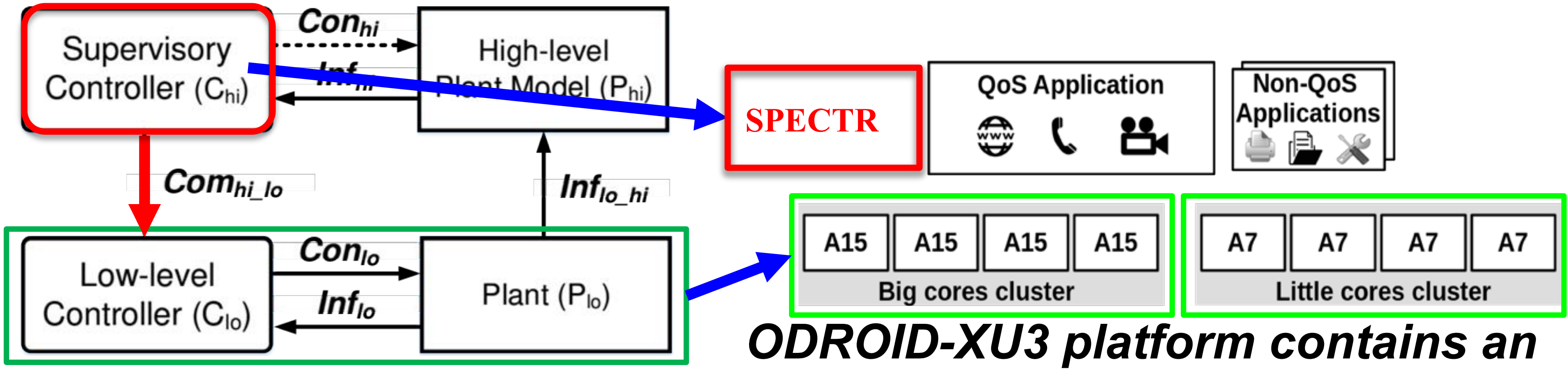
ODROID-XU3 platform contains an Exynos 5422 Octa-core SoC

*Rahmani, A. M., Donyanavard, B., Mück, T., Moazzemi, K., Jantsch, A., Mutlu, O., & Dutt, N., *SPECTR: Formal Supervisory Control and Coordination for Many-core Systems Resource Management*. ASPLOS '18

SPECTR: Our Supervisory Approach

- Autonomy and robustness through supervisory control
- Case Study*

Supervisor bounds behavior of controllers, manages goal



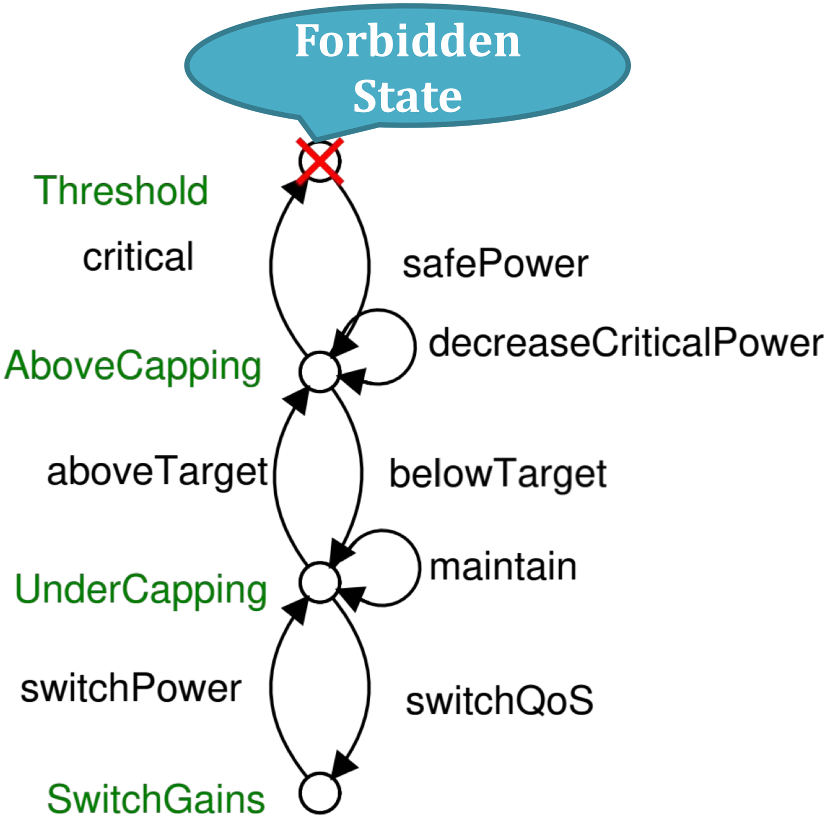
ODROID-XU3 platform contains an Exynos 5422 Octa-core SoC

Low-level controllers satisfy objective

*Rahmani, A. M., Donyanavard, B., Mück, T., Moazzemi, K., Jantsch, A., Mutlu, O., & Dutt, N., *SPECTR: Formal Supervisory Control and Coordination for Many-core Systems Resource Management*. ASPLOS '18

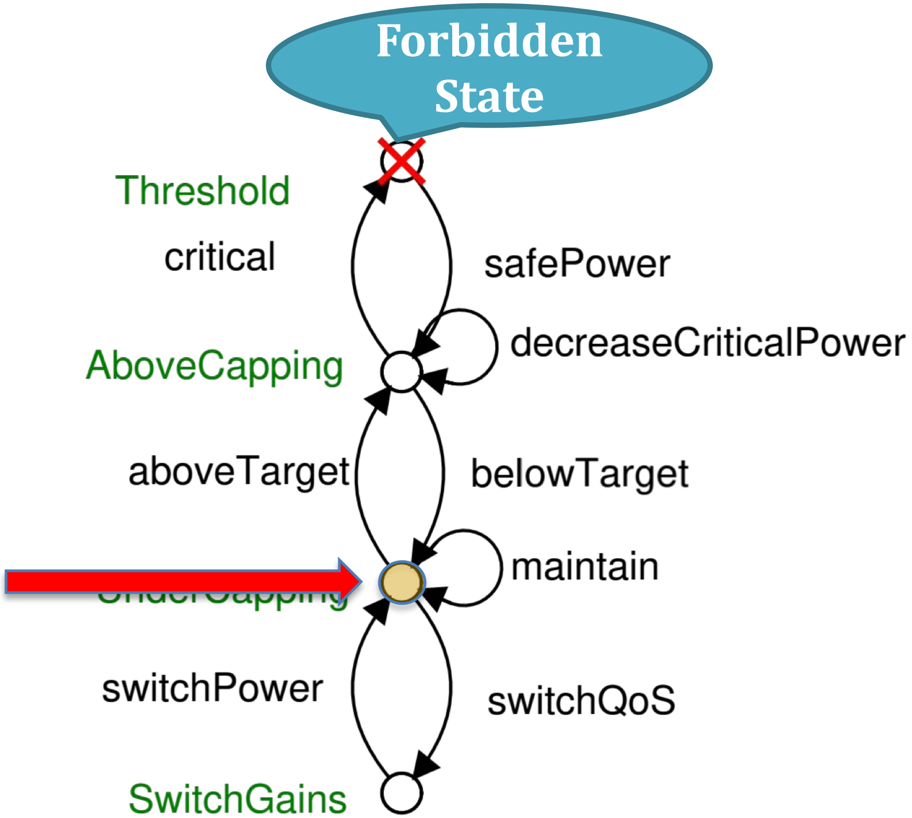
Example: Power Capping

- Specify desired behavior via accepted and forbidden states to restrict the behavior of the system



Example: Power Capping

- Specify desired behavior via accepted and forbidden states to restrict the behavior of the system



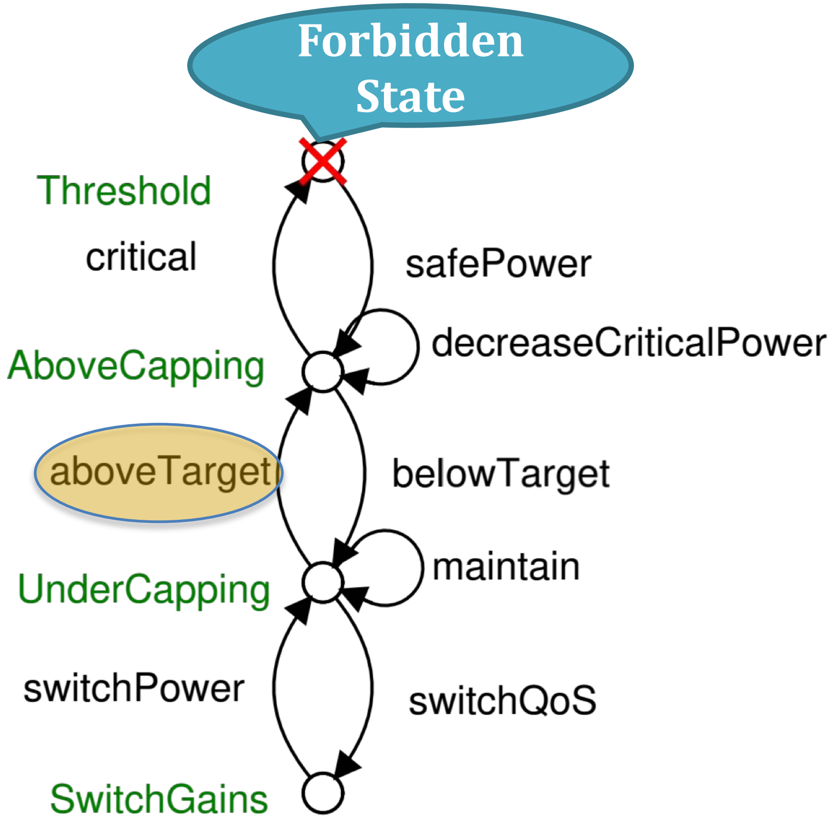
Forbidden State

If power is in safe region, prioritize QoS

Example: Power Capping

- Specify desired behavior via accepted and forbidden states to restrict the behavior of the system

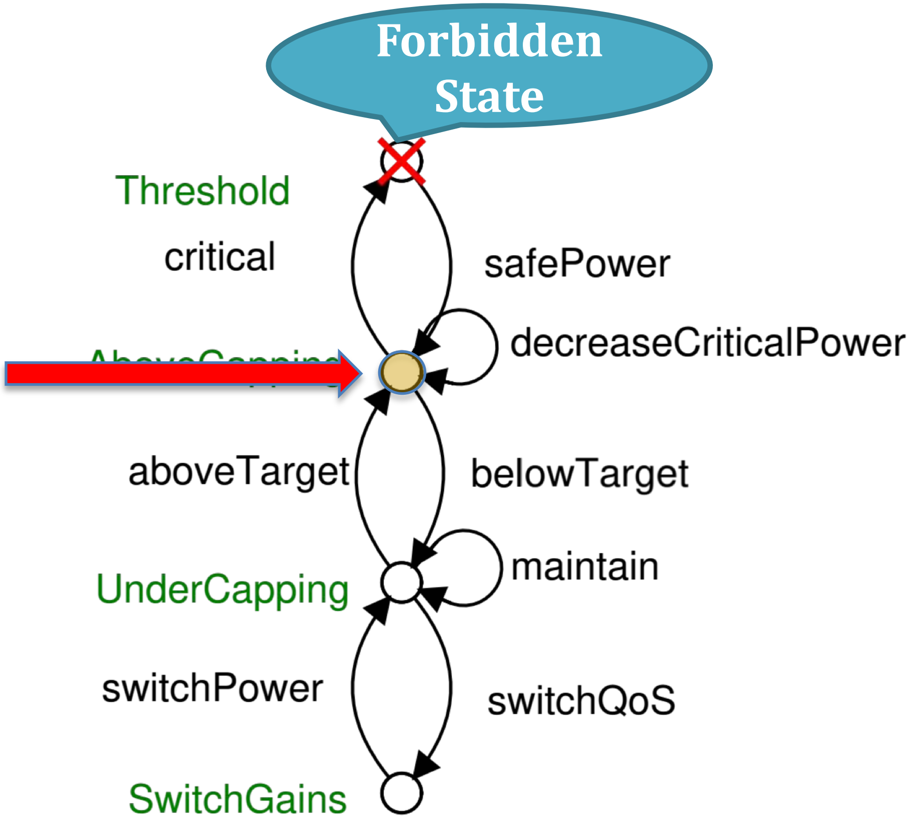
If power exceeds threshold,



Example: Power Capping

- Specify desired behavior via accepted and forbidden states to restrict the behavior of the system

If power exceeds threshold, reduce power

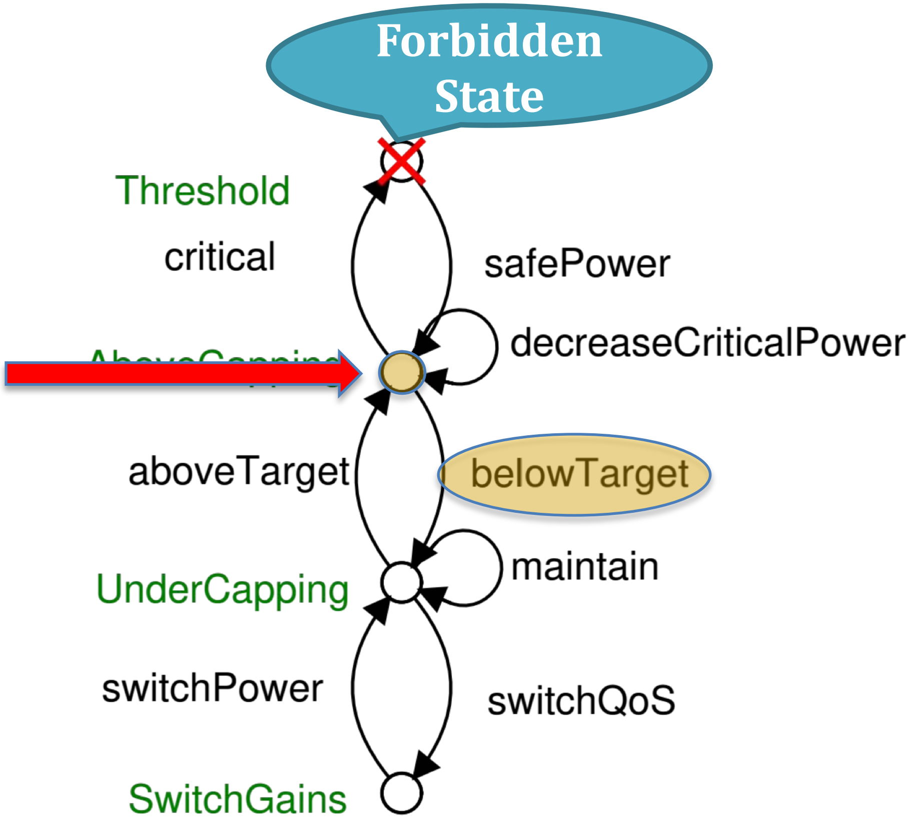


Forbidden State

Example: Power Capping

- Specify desired behavior via accepted and forbidden states to restrict the behavior of the system

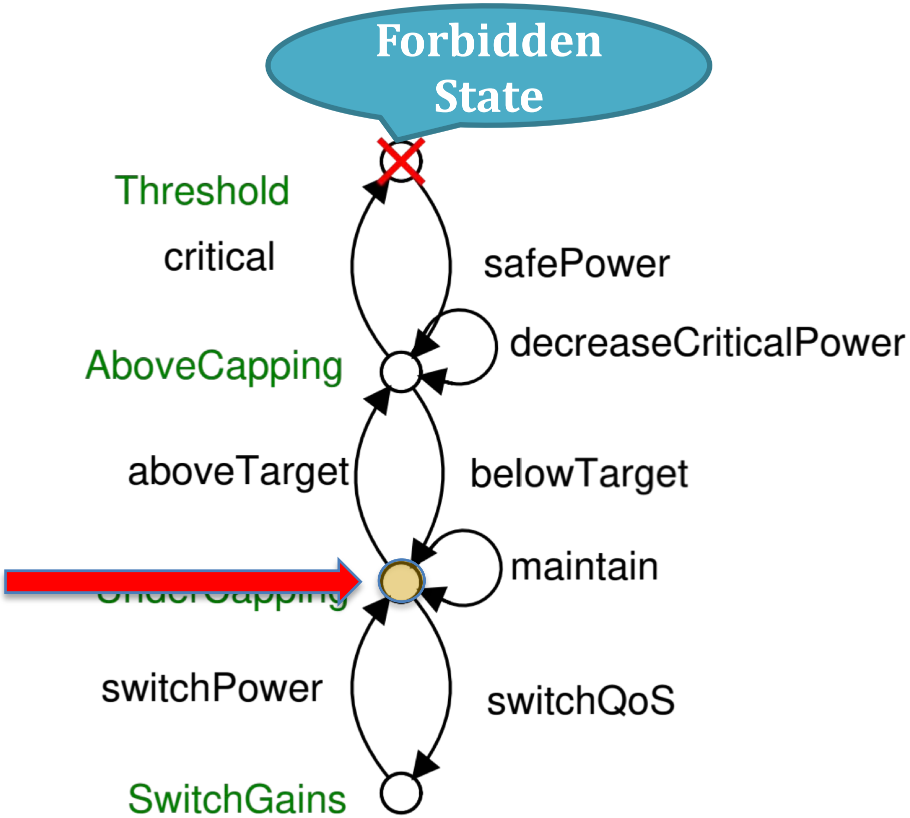
If power exceeds threshold, reduce power...until it lowers again



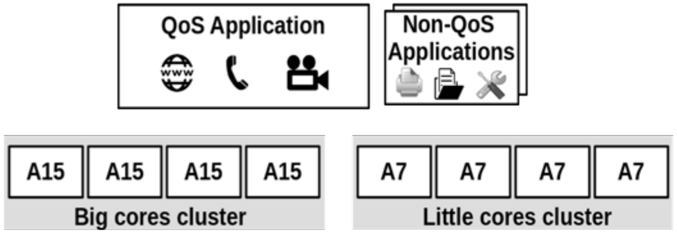
Example: Power Capping

- Specify desired behavior via accepted and forbidden states to restrict the behavior of the system

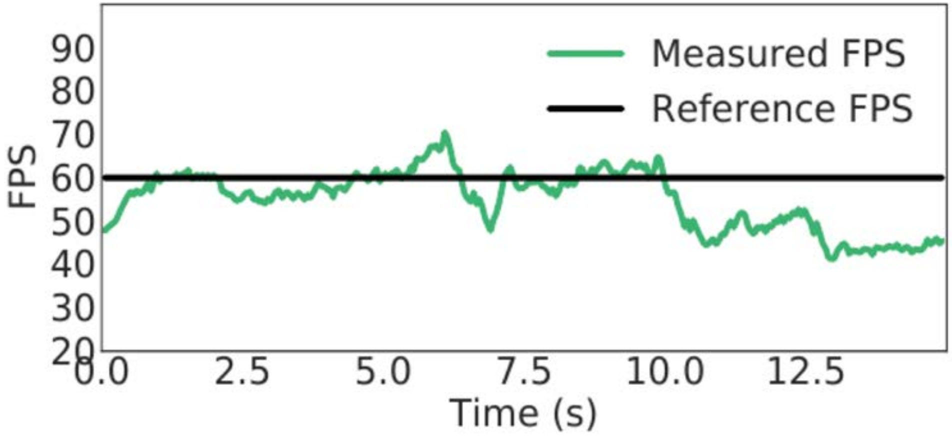
If power exceeds threshold, reduce power...until it lowers again

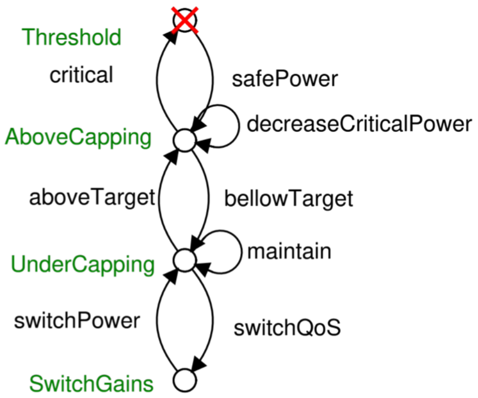


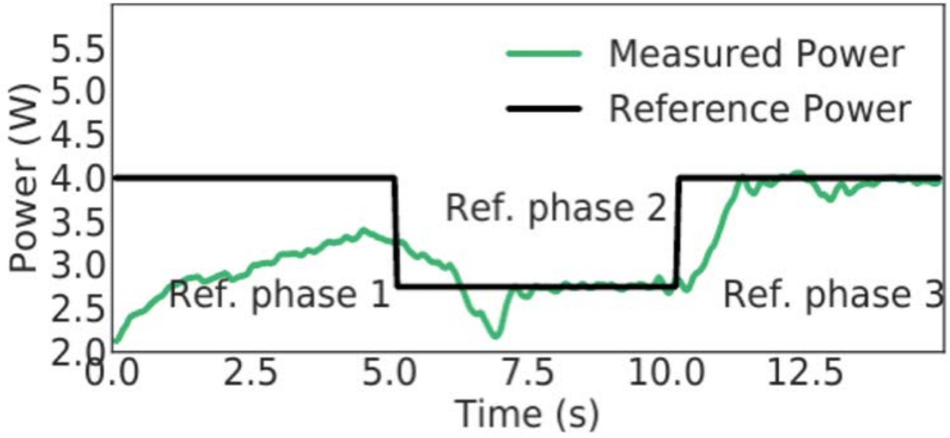
SPECTR Demonstration



QoS Task: x264

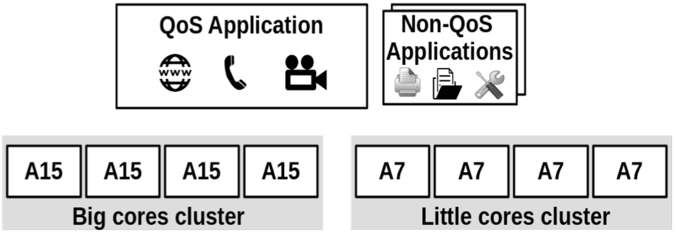






[Rahmani18] ASPLOS '18

SPECTR Demonstration



QoS Application

Non-QoS Applications

A15 A15 A15 A15

Big cores cluster

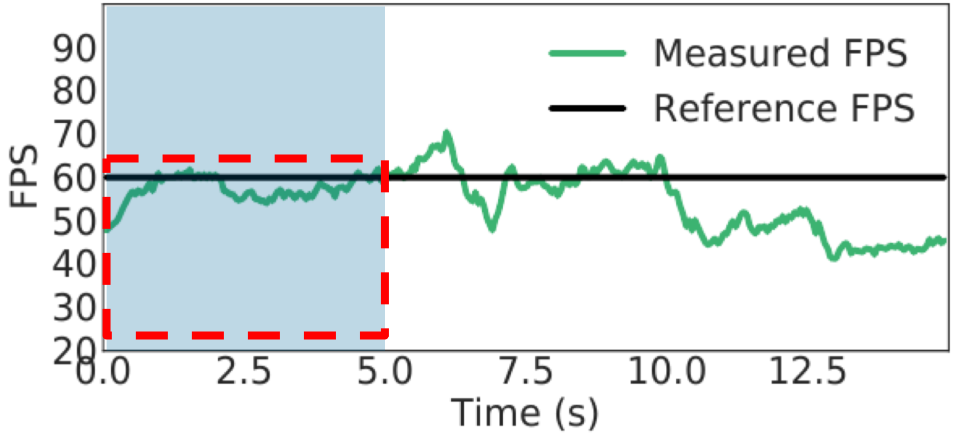
A7 A7 A7 A7

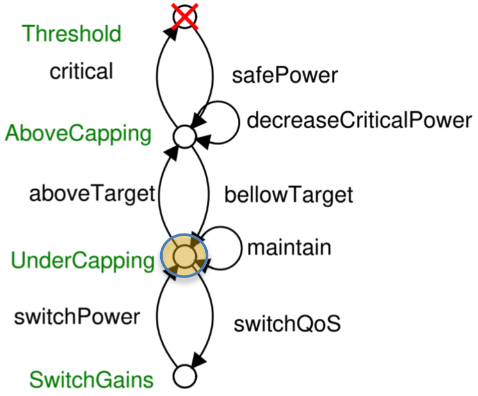
Little cores cluster

QoS Task: x264

Safe Phase: QoS app only

SPECTR satisfies FPS with minimum power





Threshold

critical

safePower

decreaseCriticalPower

AboveCapping

aboveTarget

bellowTarget

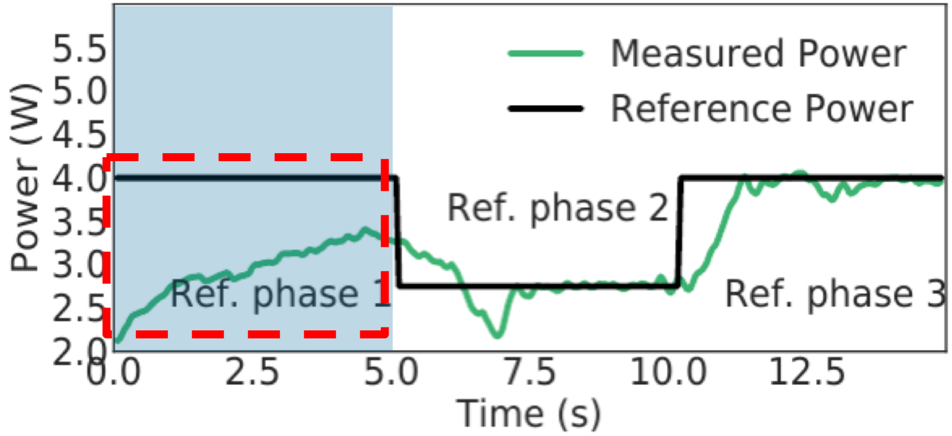
UnderCapping

maintain

switchPower

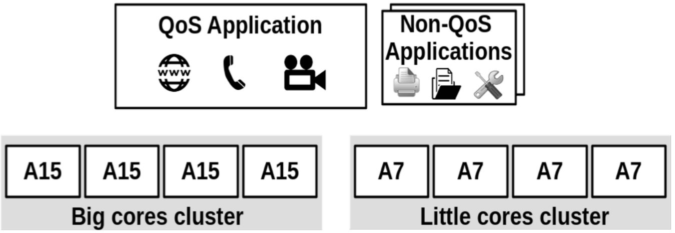
switchQoS

SwitchGains

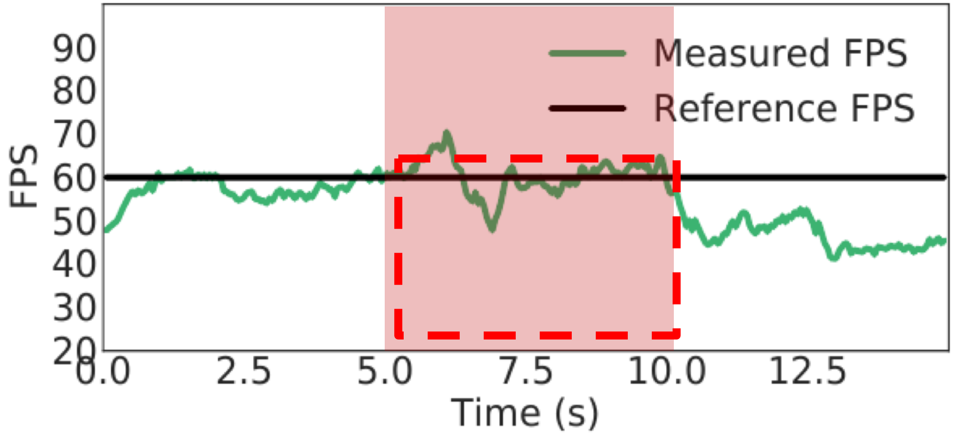


[Rahmani18] ASPLOS '18

SPECTR Demonstration

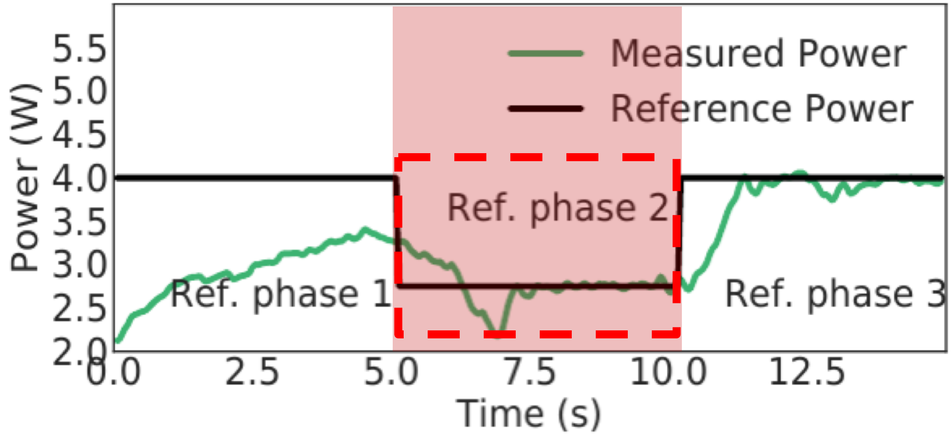


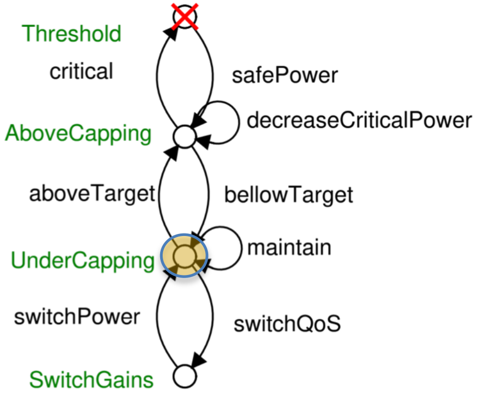
QoS Task: x264



Emergency Phase: TDP reduced in response to thermal event

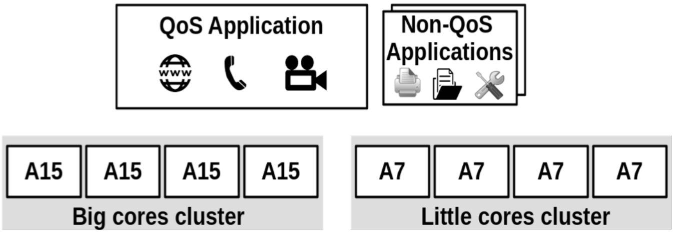
SPECTR satisfies FPS and power





[Rahmani18] ASPLOS '18

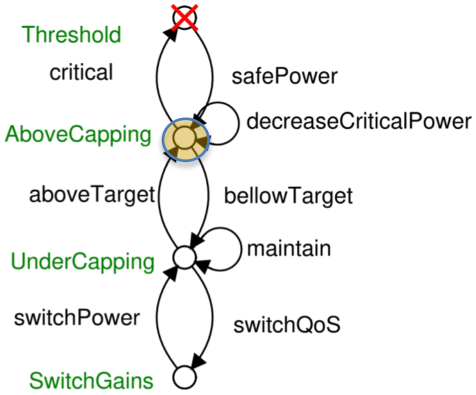
SPECTR Demonstration

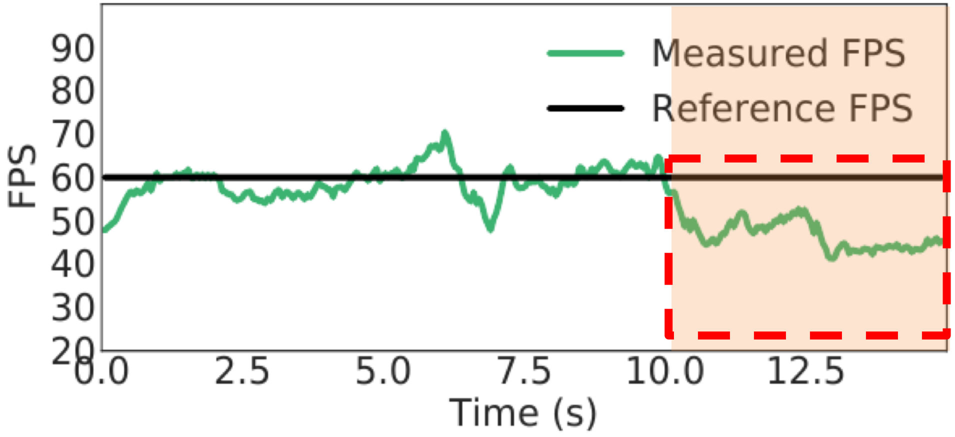


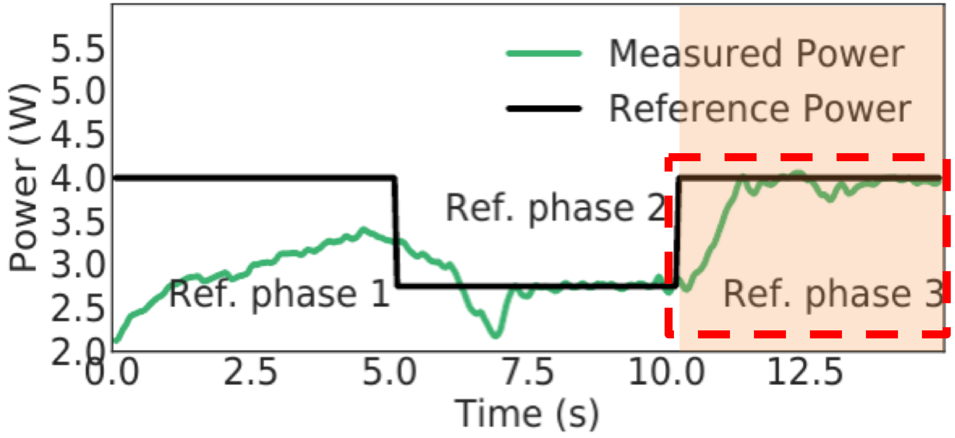
QoS Task: x264

Disturbance Phase: TDP returned to normal, background tasks introduced

SPECTR prioritizes power capping

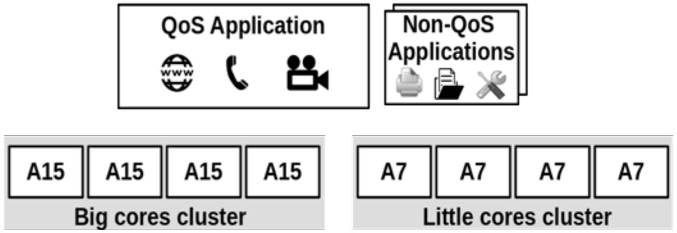






[Rahmani18] ASPLOS '18

SPECTR Demonstration



QoS Application

Non-QoS Applications

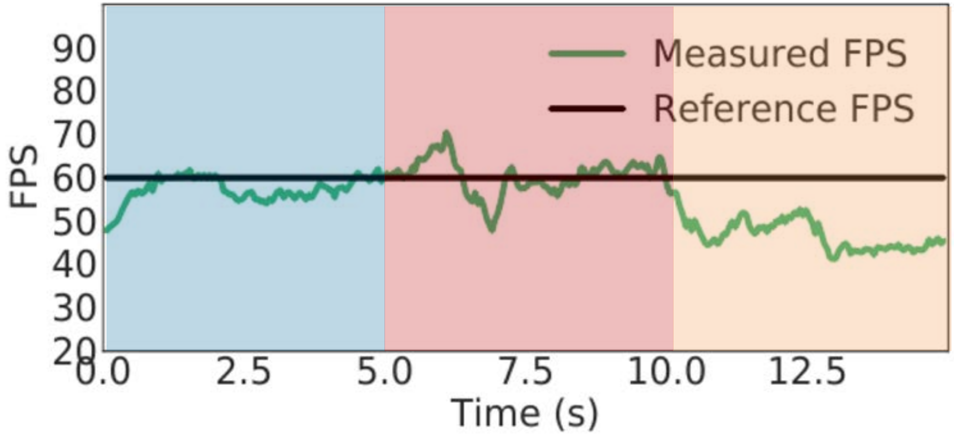
A15 A15 A15 A15

Big cores cluster

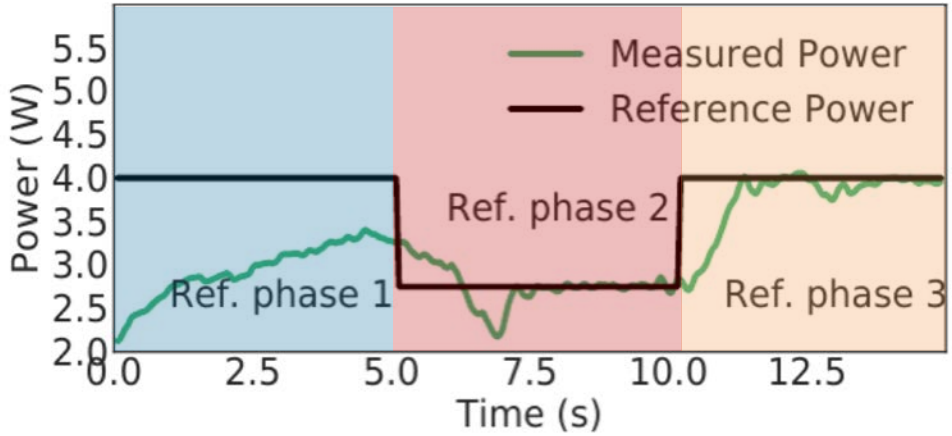
A7 A7 A7 A7

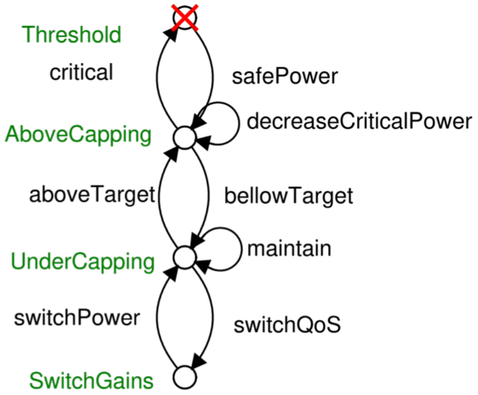
Little cores cluster

QoS Task: x264



SPECTR meets FPS target when possible, while honoring power cap





Threshold critical

safePower

AboveCapping

decreaseCriticalPower

aboveTarget

bellowTarget

UnderCapping

maintain

switchPower

switchQoS

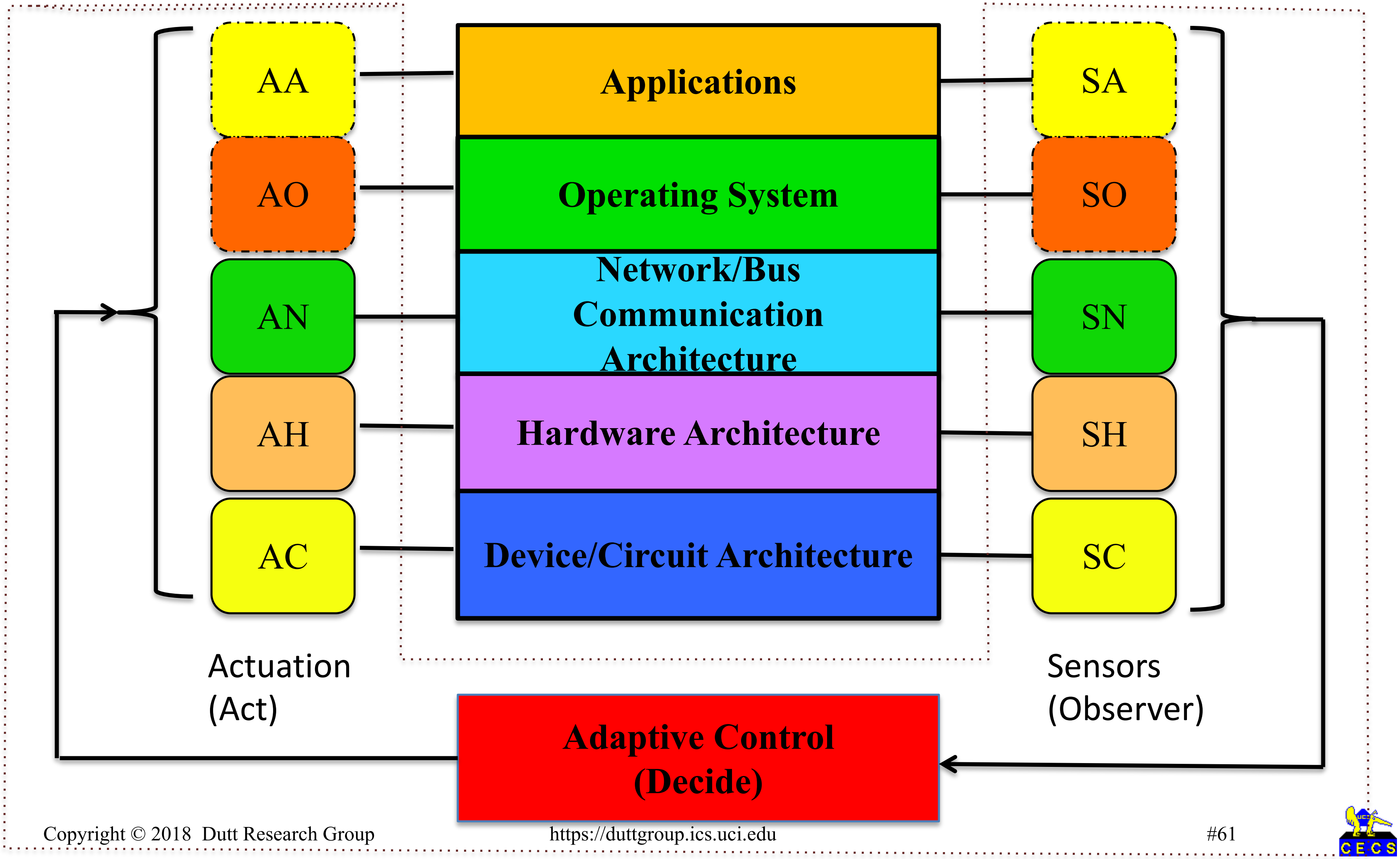
SwitchGains

[Rahmani18] ASPLOS '18

Outline

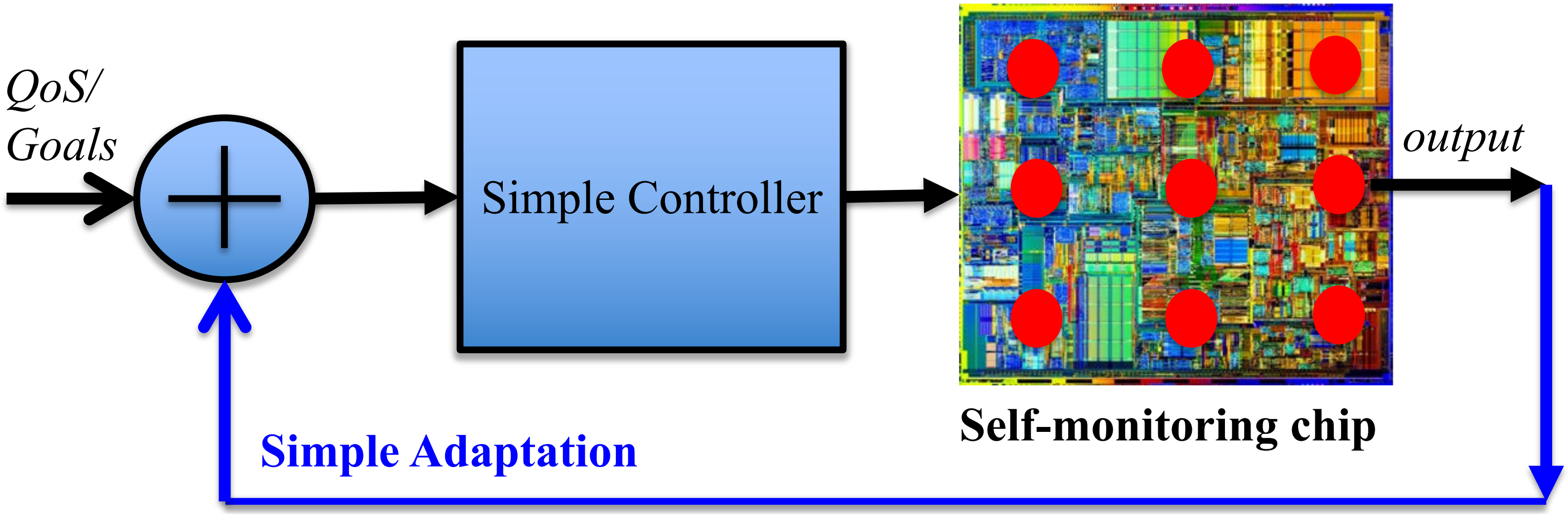
- Computational Self-Awareness
- Why Self-Aware Chips?
- Cross-Layer Sensing & Actuation
- Towards Self-Aware Chips
- Supervisory Control & Coordination
- Wrap-up

Key Take-Away 1: Cross-Layer Physical/Virtual Sensing & Actuation



From today's chips

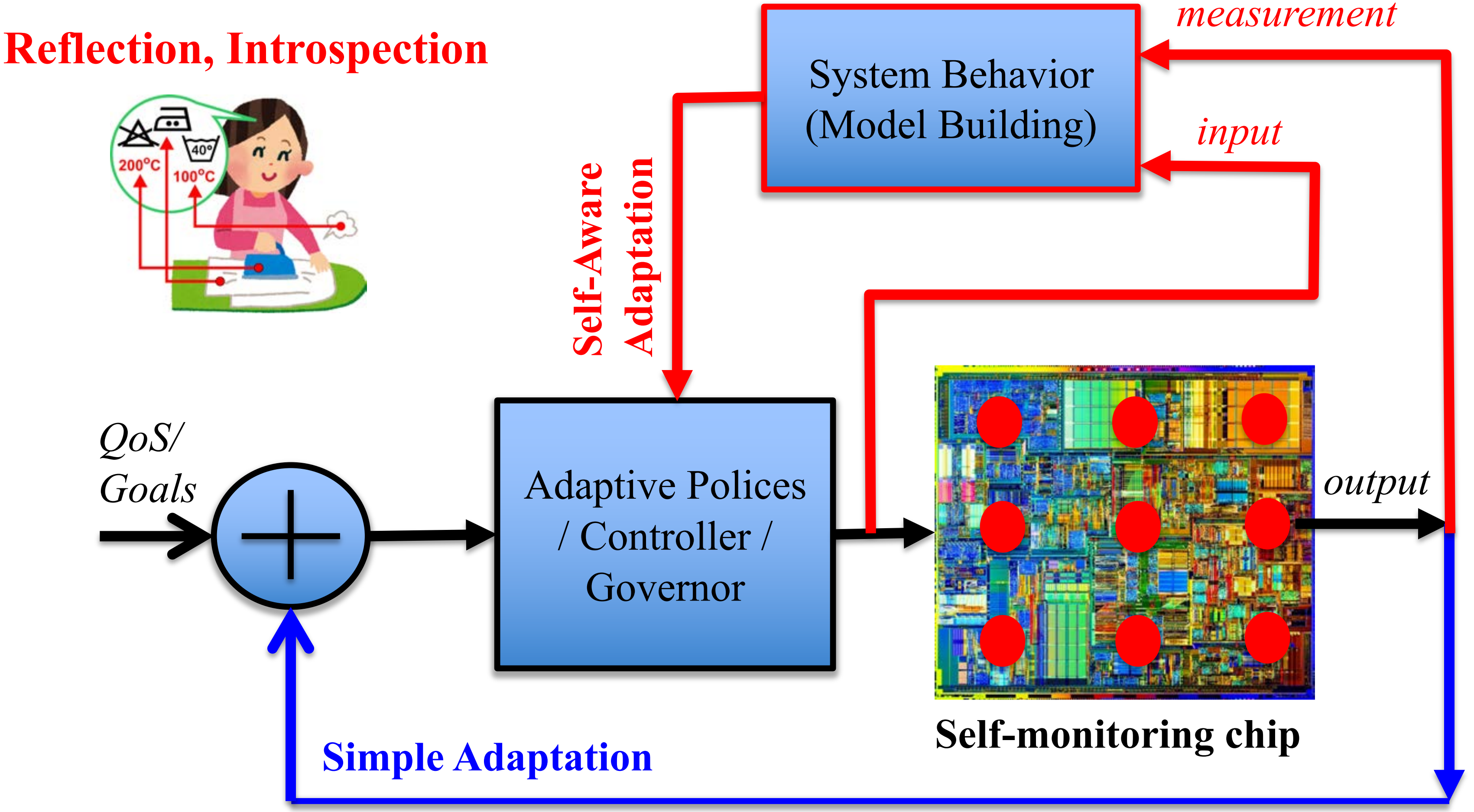
Reflexive, Reactive



Self-monitoring and **simple adaptation**

Key Take-Away 2:

Towards on-chip self-awareness



Reflection, Introspection

measurement

System Behavior (Model Building)

input

Self-Aware Adaptation

QoS/ Goals

Adaptive Polices / Controller / Governor

output

Self-monitoring chip

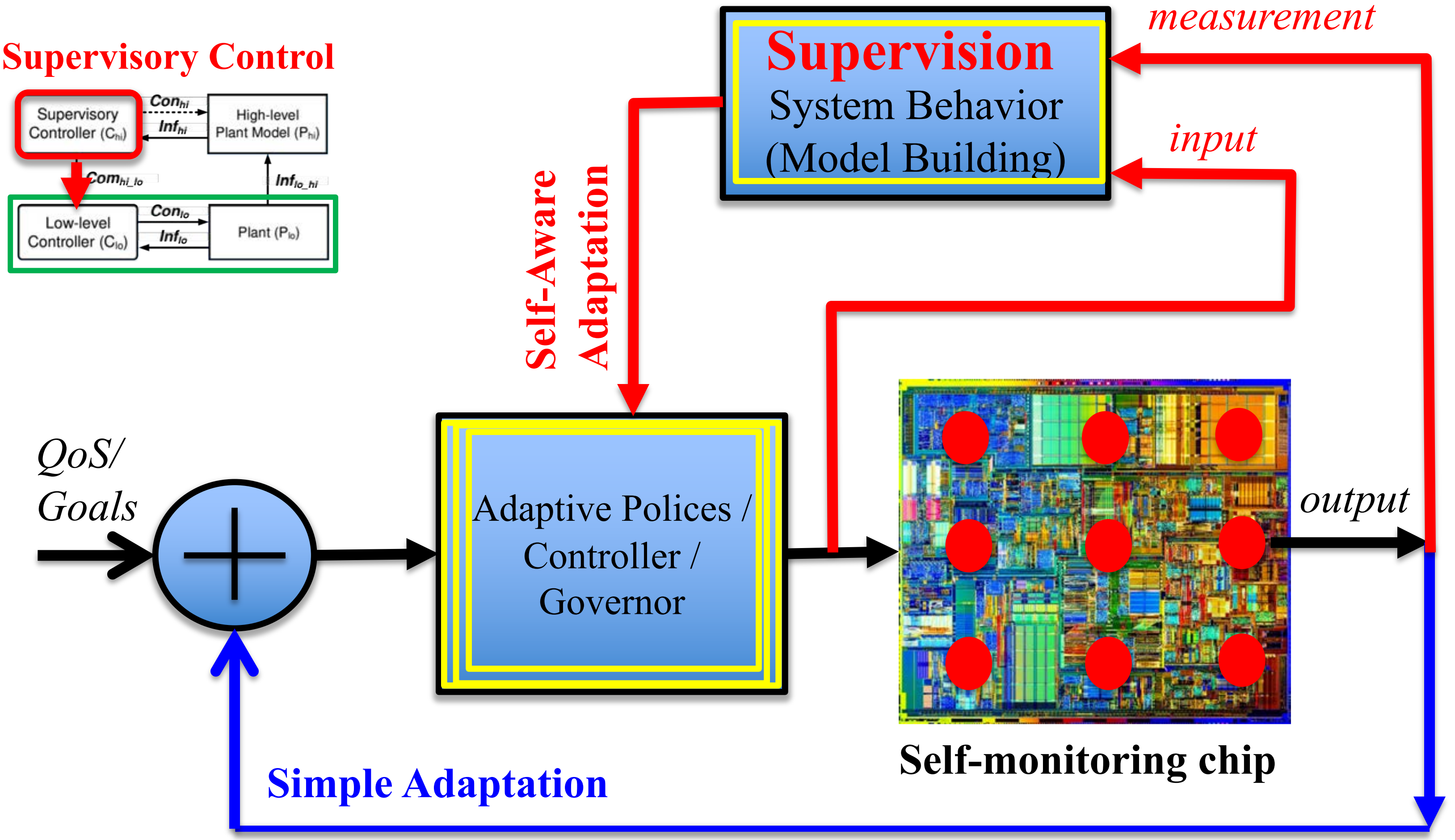
Simple Adaptation

Self-monitoring and **Self-modeling**

[Sarma14, CODES+ISSS14]

Key Take-Away 3:

Supervisory Control & Coordination



Supervisory Control & Coordination

[Rahmani18, ASPLOS 2018]

IEEE DESIGN & TEST November/December 2017

Special Issue on Self-Awareness in Systems on Chip 2017

VOLUME 34 NUMBER 6

IEEE

Design&Test

NOVEMBER/DECEMBER 2017

Special Issue on Self-Awareness in Systems on Chip 2017

- Self-Awareness in Systems on Chip—A Survey
- Health Management for Self-Aware SoCs Based on IEEE 1687 Infrastructure
- KOCL: Power Self-Awareness for Arbitrary FPGA-SoC-Accelerated OpenCL Applications
- A Self-Aware Architecture for PVT Compensation and Power Nap in Near-Threshold Processors
- Self-Adaptive Timing Repair

Contents

November/December 2017
Volume 34 Number 6

Copublished by the IEEE Council on Electronic Design Automation, the IEEE Circuits and Systems Society, the IEEE Solid-State Circuits Society, and the Test Technology Technical Council

Special Issue

6 **Guest Editorial: Special Issue on Self-Aware Systems on Chip**
Axel Jantsch and Nikil Dutt

8 **Self-Awareness in Systems on Chip—A Survey**
Axel Jantsch, Nikil Dutt, and Amir M. Rahmani

27 **Health Management for Self-Aware SoCs Based on IEEE 1687 Infrastructure**
Konstantin Shibin, Sergei Devadze, Artur Jutman, Martin Grabmann, and Robin Pricken

36 **KOCL: Power Self-Awareness for Arbitrary FPGA-SoC-Accelerated OpenCL Applications**
James J. Davis, Joshua M. Levine, Edward A. Stott, Eddie Hung, Peter Y. K. Cheung, and George A. Constantinides

46 **A Self-Aware Architecture for PVT Compensation and Power Nap in Near-Threshold Processors**
Davide Rossi, Igor Loi, Antonio Pullini, Christoph Müller, Andreas Burg, Francesco Conti, Luca Benini, and Philippe Flatresse

54 **Self-Adaptive Timing Repair**
Hans Giesen, Raphael Rubin, Benjamin Gojman, and André DeHon

Survey Paper

63 **Trusted Analog/Mixed-Signal/RF ICs: A Survey and a Perspective**
Angelos Antonopoulos, Christiana Kapatsori, and Yiorgos Makris

General Interest

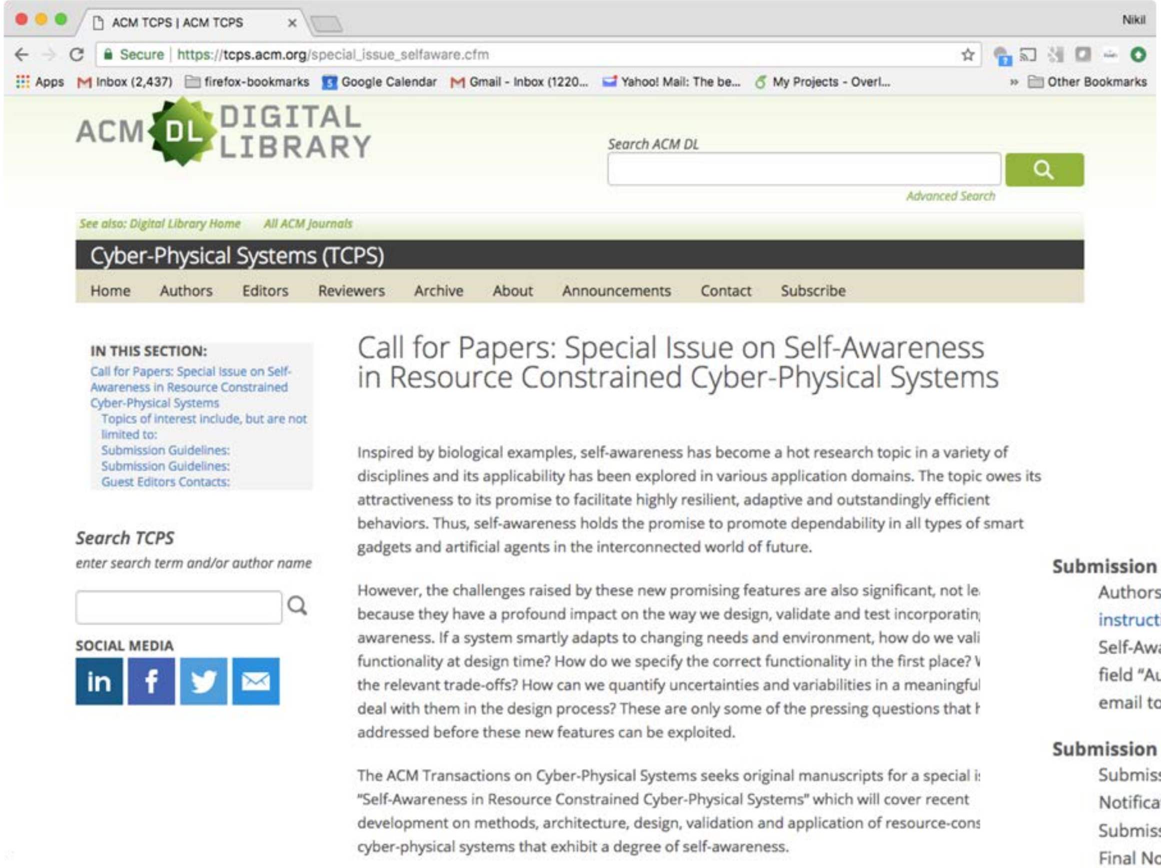
77 **Layout-Aware Optimized Prebond Silicon Interposer Test Synthesis**
Katherine Shu-Min Li, Sying-Jyan Wang, Ruei-Ting Gu, and Bo-Chuan Cheng

84 **Adaptive ECC for Tailored Protection of Nanoscale Memory**
Dongyeob Shin, Jongsun Park, Jangwon Park, Somnath Paul, and Swarup Bhunia

94 **Low-Power Sparse Hyperdimensional Encoder for Language Recognition**
Mohsen Imani, John Hwang, Tajana Rosing, Abbas Rahimi, and Jan M. Rabaey

ISSN: 2168-2356

SelPhyS 2019 and TCPS Special Issue



IN THIS SECTION:

- Call for Papers: Special Issue on Self-Awareness in Resource Constrained Cyber-Physical Systems
  - Topics of interest include, but are not limited to:
  - Submission Guidelines:
  - Submission Guidelines:
  - Guest Editors Contacts:

Search TCPS

enter search term and/or author name

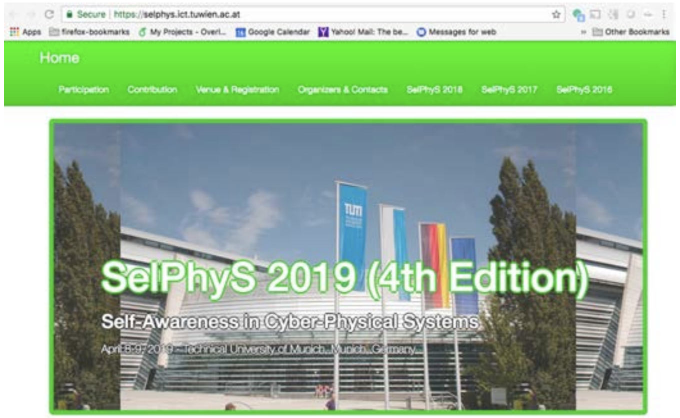
SOCIAL MEDIA

Call for Papers: Special Issue on Self-Awareness in Resource Constrained Cyber-Physical Systems

Inspired by biological examples, self-awareness has become a hot research topic in a variety of disciplines and its applicability has been explored in various application domains. The topic owes its attractiveness to its promise to facilitate highly resilient, adaptive and outstandingly efficient behaviors. Thus, self-awareness holds the promise to promote dependability in all types of smart gadgets and artificial agents in the interconnected world of future.

However, the challenges raised by these new promising features are also significant, not le
because they have a profound impact on the way we design, validate and test incorporatin
awareness. If a system smartly adapts to changing needs and environment, how do we vali
functionality at design time? How do we specify the correct functionality in the first place? V
the relevant trade-offs? How can we quantify uncertainties and variabilities in a meaningful
deal with them in the design process? These are only some of the pressing questions that h
addressed before these new features can be exploited.

The ACM Transactions on Cyber-Physical Systems seeks original manuscripts for a special i
"Self-Awareness in Resource Constrained Cyber-Physical Systems" which will cover recent
development on methods, architecture, design, validation and application of resource-cons
cyber-physical systems that exhibit a degree of self-awareness.



Submission Guidelines:

Authors should submit their journal version at Manuscript Central adhering to the formatting instructions on the TCPS Web page, and indicate that you are submitting to the Special Issue on Self-Awareness in Resource Constrained Cyber-Physical Systems" on the first page and in the field "Author's Cover Letter:" in Manuscript Central). For additional questions, please send an email to any of the guest editors: p.lewis@aston.ac.uk, axel.jantsch@tuwien.ac.at, dutt@uci.edu.

Submission Guidelines:

Submission deadline: 7 September, 2018
Notification of First Round: 7 December, 2018
Submission of Revision: 8 February, 2019
Final Notification: 12 April 2019
Final Paper Due: 23 May 2019

Guest Editors Contacts:

Peter Lewis, p.lewis@aston.ac.uk
Axel Jantsch, axel.jantsch@tuwien.ac.at
Nikil Dutt, dutt@uci.edu

All ACM Journals | *See Full Journal Index*

Ongoing Efforts

- More heterogeneity (CPU+GPU+DSP+NPU+FPGA+…..)
  - Reconfigure workloads at runtime to freely migrate between resources
  - Complex predictive models
- Distributed management
  - Propagating sensing info across non-coherent processing units
- Non-compute resources
  - Memory and I/O

Ongoing Challenges

- Self-trained models
  - Add feedback for error correction
  - Challenging for models that are non-linear and/or based on heuristics
- Machine learning
  - Replacement for analytical/heuristic-based models ?
  - Unsupervised machine learning to mine sensing data and find patterns for optimizing policies or creating new ones
- Policy supervisors
  - Provide formal or stronger guarantees

Acknowledgements

- Dutt Research Group
  - **Tiago Mück**, Amir Rahmani, Santanu Sarma, Majid Shoushtari, Bryan Donyanavard, Kasra Moazzemi, Roger Hsieh, Jurngyu Park, Hossein Tajik, Biswadip Maity
- Collaborating Faculty:
  - UCI: Fadi Kurdahi
  - Austria: TU Wien: Prof. Axel Jantsch
  - Germany: TUM: Prof. Andreas Herkersdorf, TUB: Prof. Rolf Ernst
- NSF Information Processing Factory (IPF) project

Questions?

Dutt Research Group: http://duttgroup.ics.uci.edu/