

RESEARCH REVIEW 2019

Using All Processor Cores While
Being Confident about Timing

Dr. Bjorn Andersson

Copyright 2019 Carnegie Mellon University.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

Carnegie Mellon® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

DM19-1115

Complex, cyber-physical DoD systems depend on correct timing—any timing failure could be disastrous. What's more, while these systems drive demand for use of multicore processors, concern about timing has led to disabling all processor cores except one—limiting system capability.

We aim to develop a solution to overcome this obstacle.

DoD Systems Interact with Their Physical Environment



DoD Systems Include Software



DoD Systems Include Software That Interacts with the Physical Environment



DoD Systems Include Software That Has Real-Time Requirements



Satisfying Real-Time Requirements Is a Challenge for the DoD in General



Satisfying Real-Time Requirements Is Challenging for Upgrading the Blackhawk UH-60 Helicopter



Satisfying Real-Time Requirements Is Challenging for Upgrading the Blackhawk UH-60 Helicopter

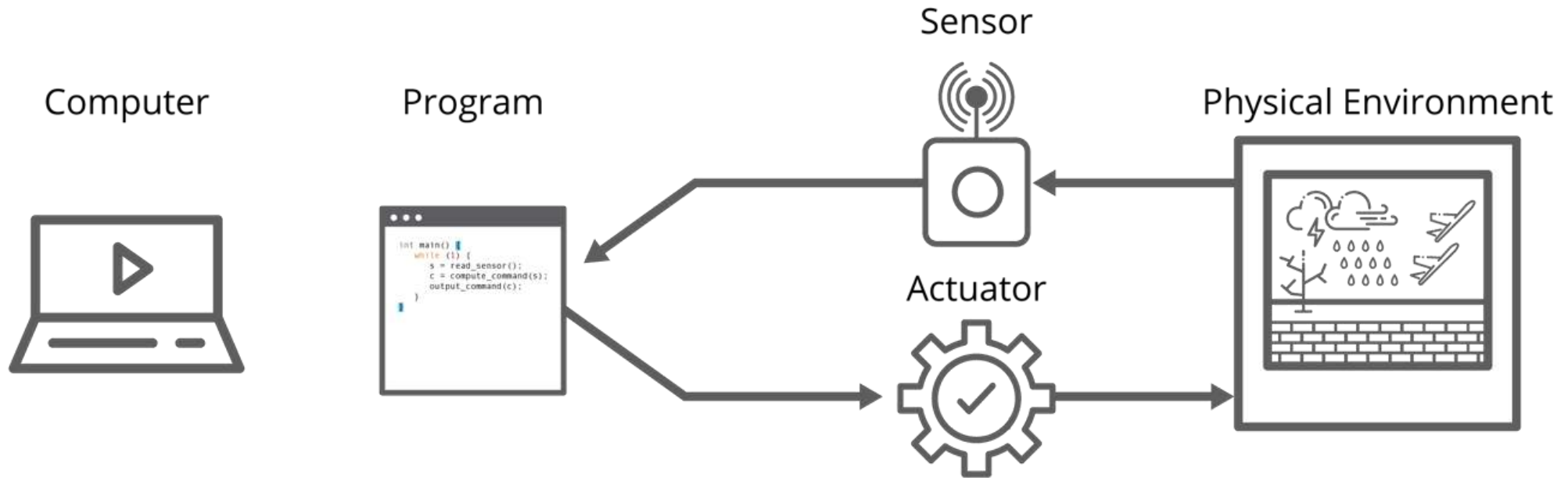


“The trick there, when you’re processing flight critical information, it has to be a deterministic environment, meaning we know exactly where a piece of data is going to be exactly when we need to — no room for error,” Langhout says. “On a multi-core processor there’s a lot of sharing going on across the cores, so right now we’re not able to do that.”

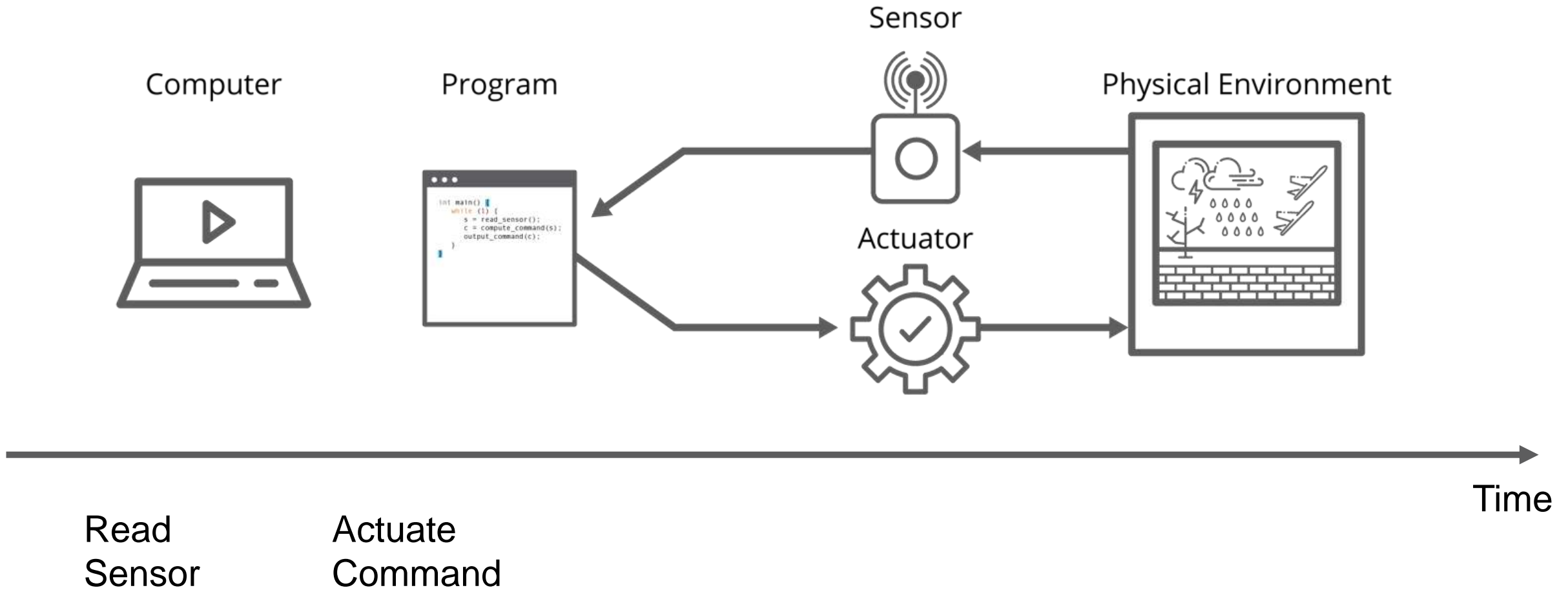
- Jeff Langhout, Acting Director, U.S. Army Aviation and Missile Research Development and Engineering Center (AMRDEC)

Source: “Army still working on multi-core processor for UH-60V,” *FlightGlobal*, May 2017.
<https://www.flightglobal.com/news/articles/army-still-working-on-multi-core-processor-for-uh-6-436895/>

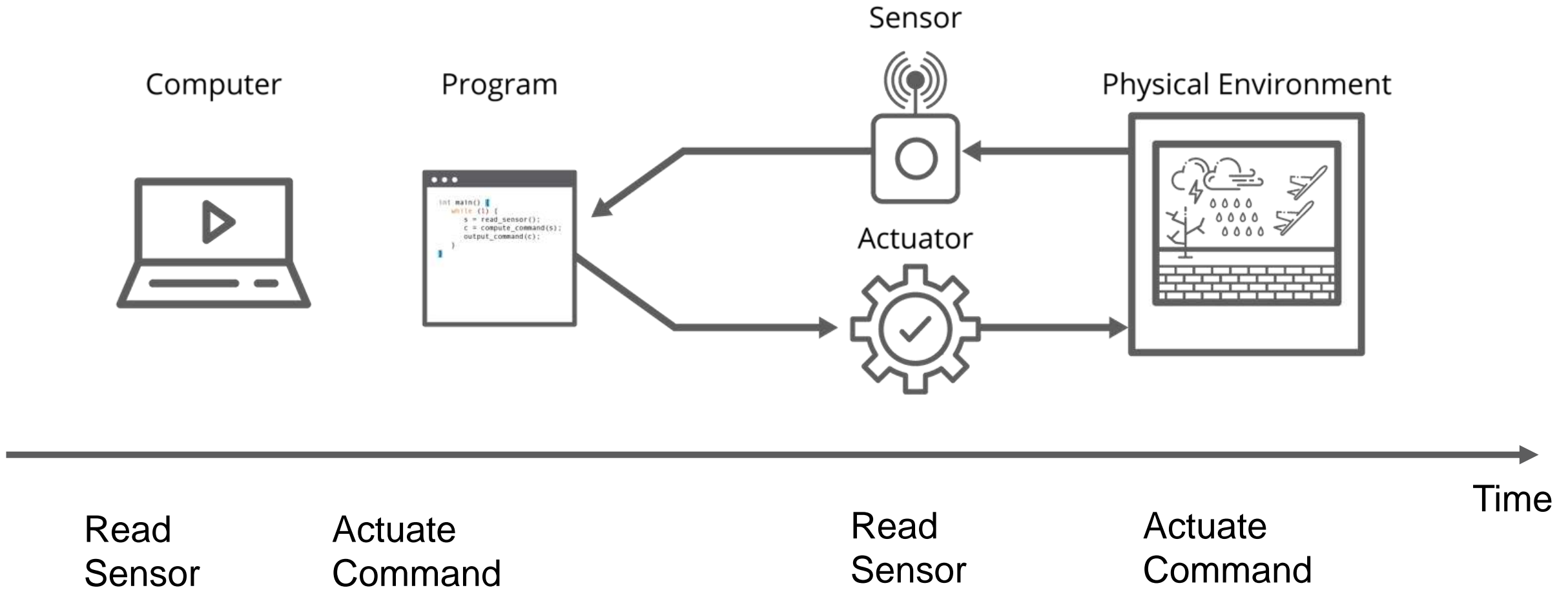
Commonality of DoD Systems



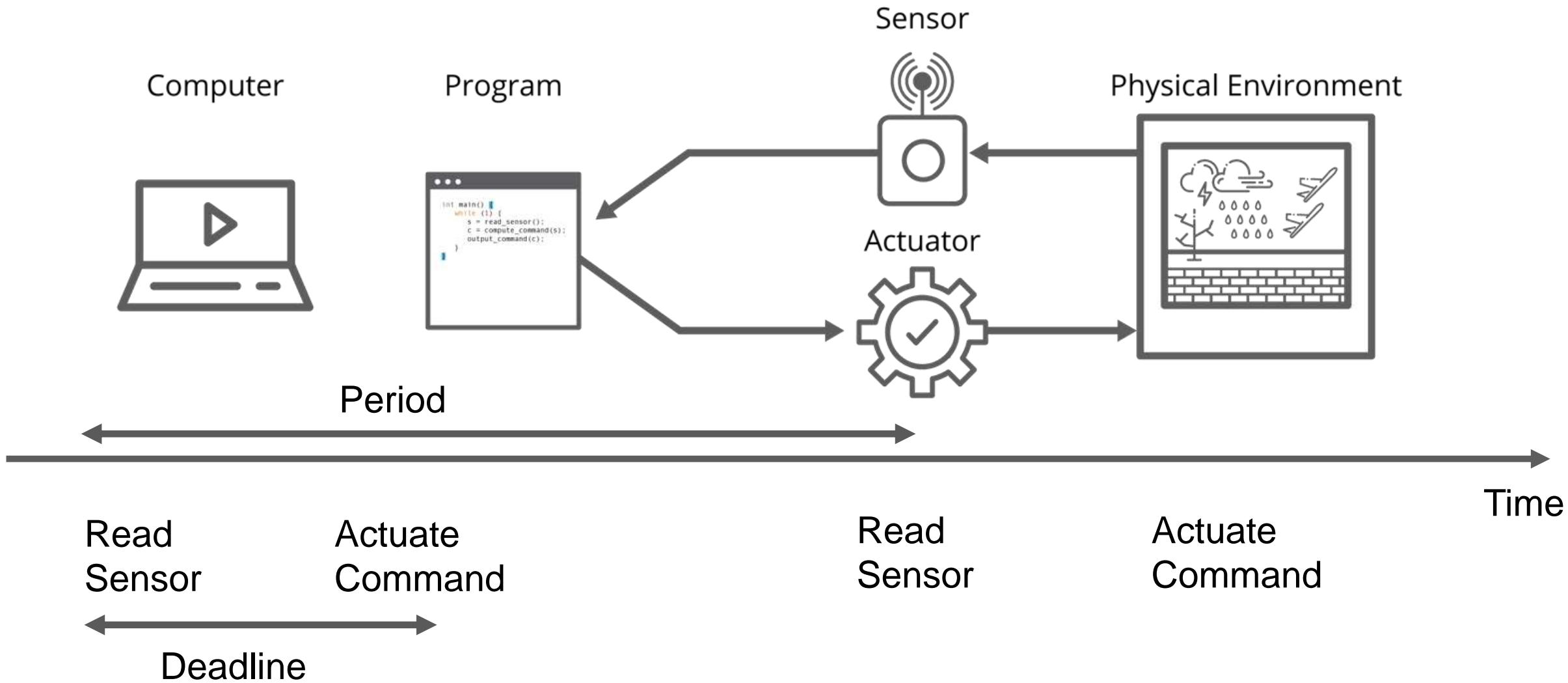
Commonality of DoD Systems



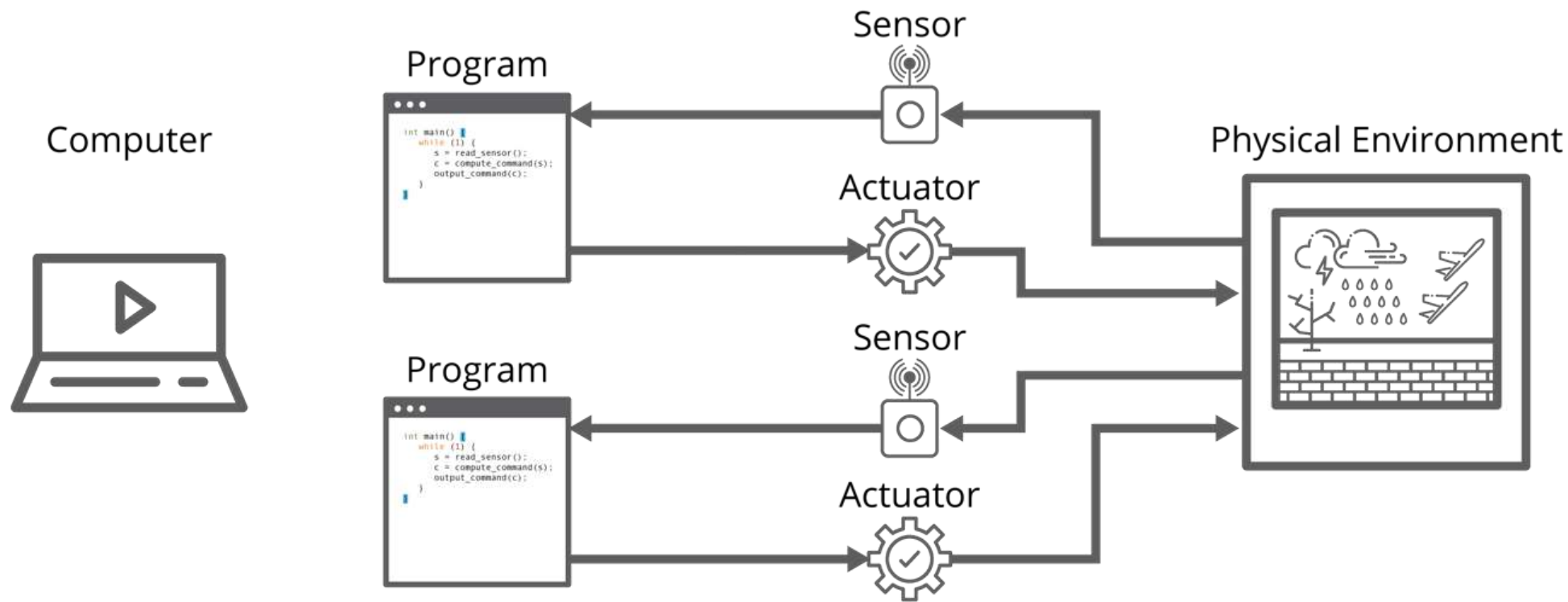
Commonality of DoD Systems



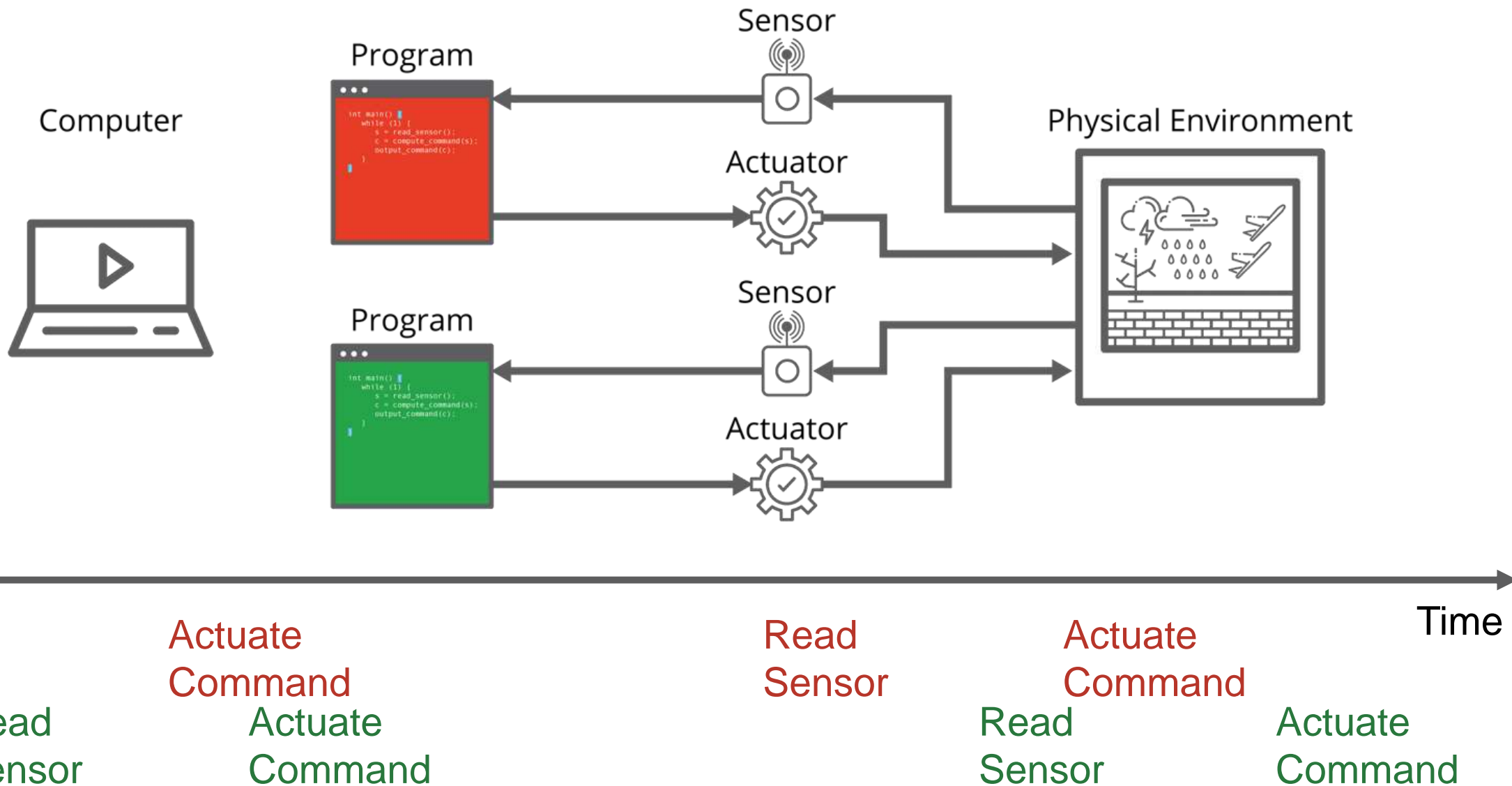
Commonality of DoD Systems



Commonality of DoD Systems



Commonality of DoD Systems



What Makes It Challenging to Satisfy Real-Time Requirements?

What Causes Delay of Software?

What Causes Delay of Software?



Time

What Causes Delay of Software?



What Causes Delay of Software?

Thread executes
one path



Time when one thread
in the software system arrives

Deadline Time

What Causes Delay of Software?

Thread executes another path



Time when one thread
in the software system arrives

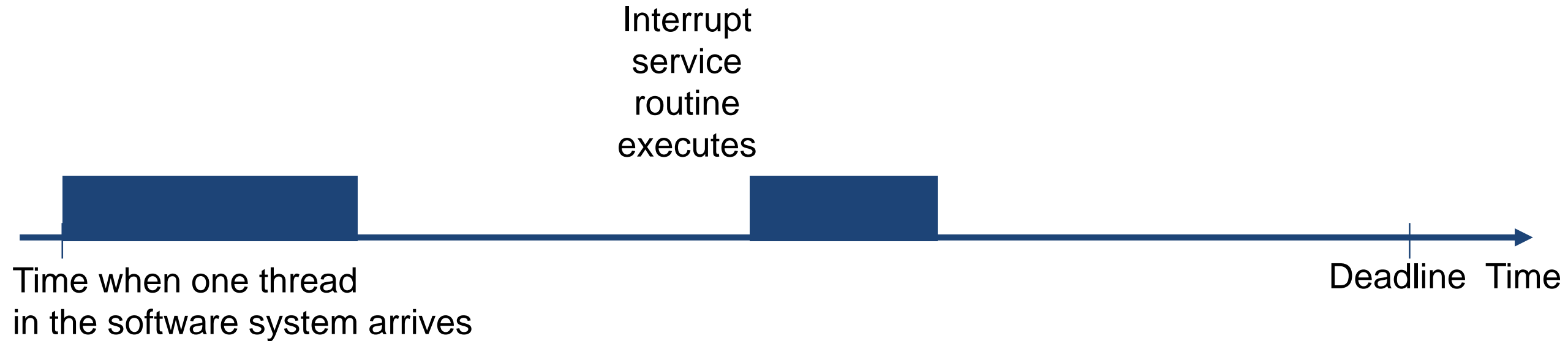
Deadline Time

What Causes Delay of Software?

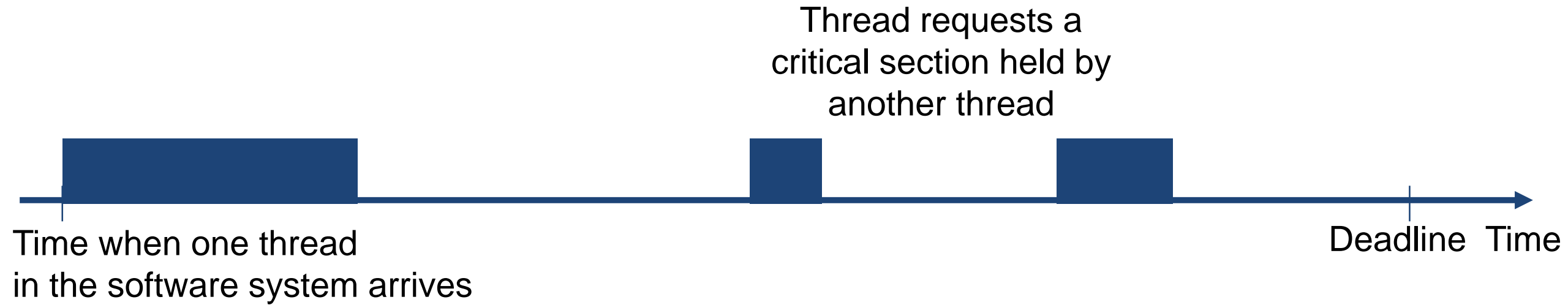
Preemption:
Another thread uses
the processor.



What Causes Delay of Software?



What Causes Delay of Software?



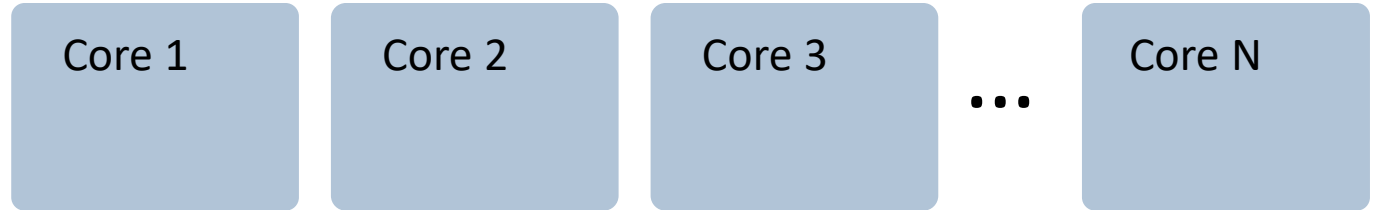
What Causes Delay of Software?



Real-Time Requirements of Software Executing on a Multicore Processor

Hardware Trends

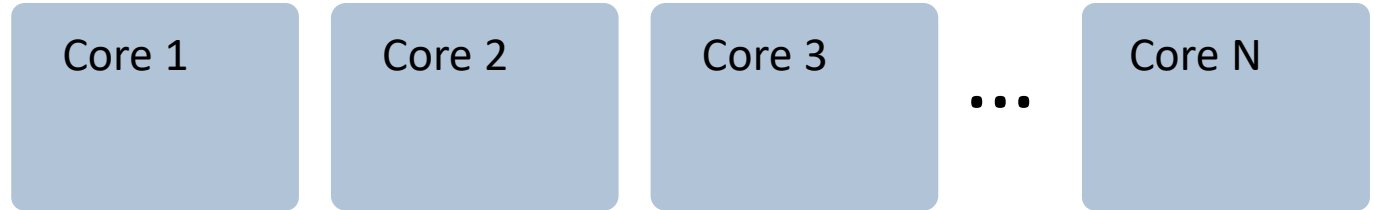
- *All computers are multicores.*



Real-Time Requirements of Software Executing on a Multicore Processor

Hardware Trends

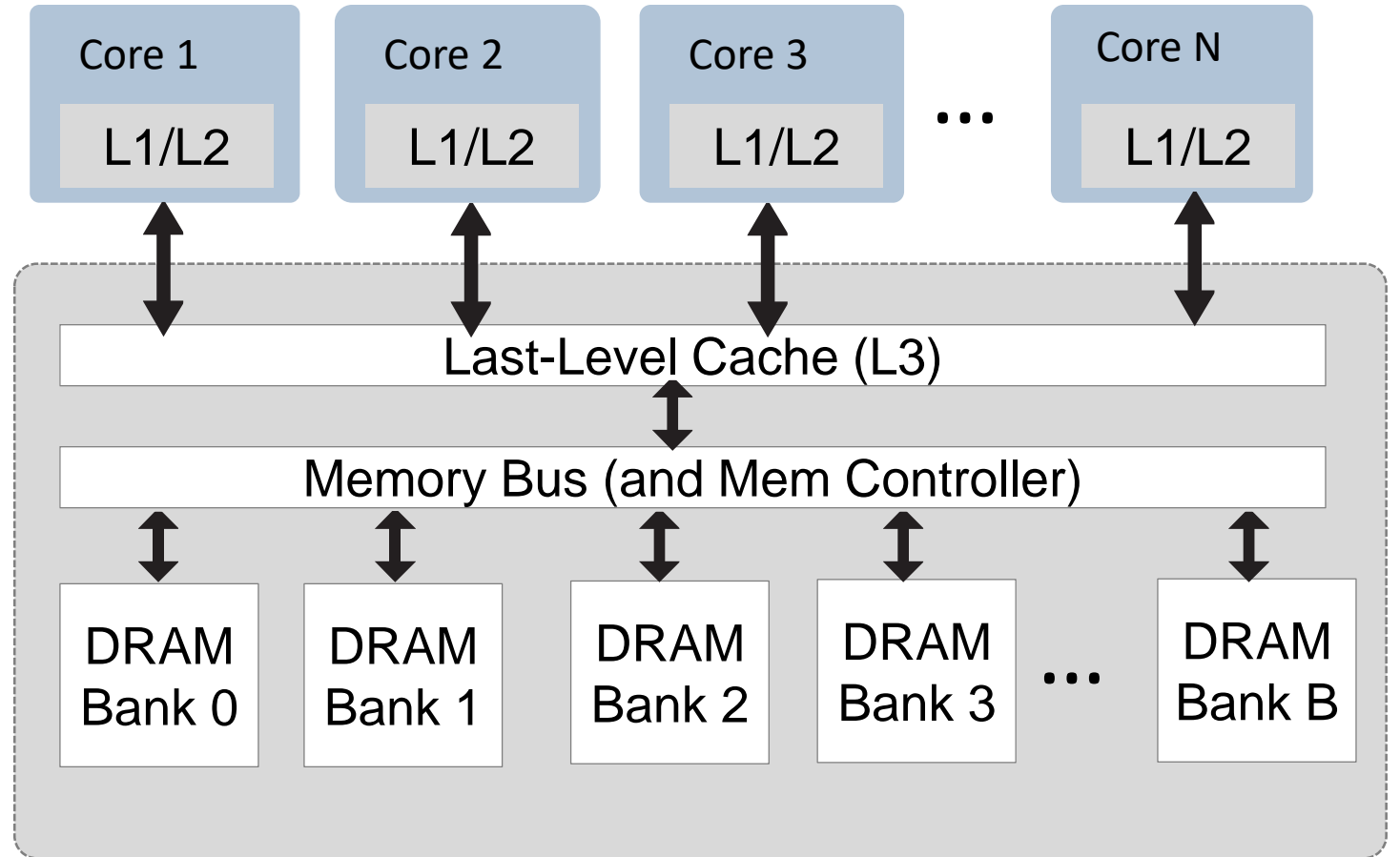
- *All computers are multicores.*
- *Most chip makers do not offer single core.*



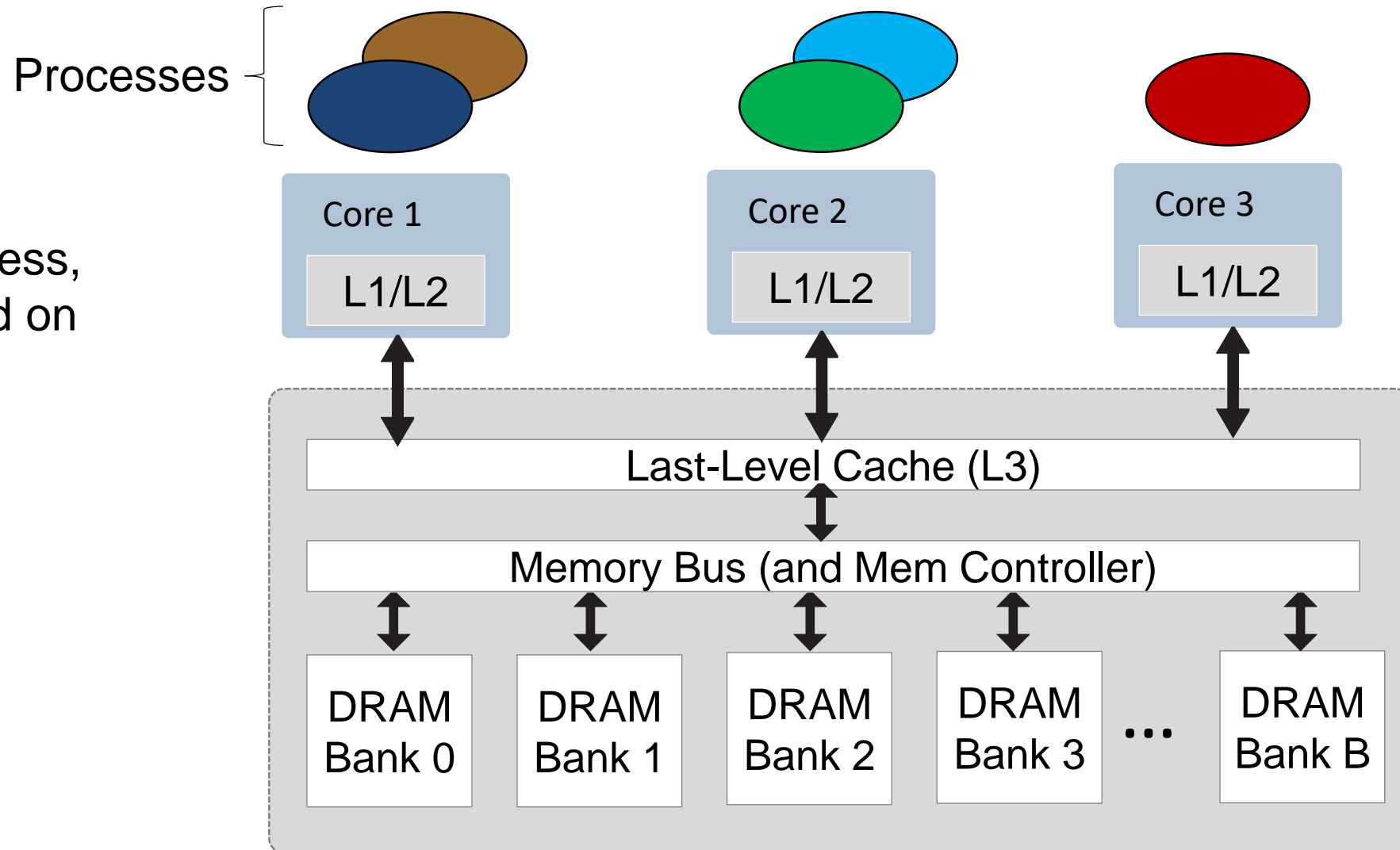
Real-Time Requirements of Software Executing on a Multicore Processor

Hardware Trends

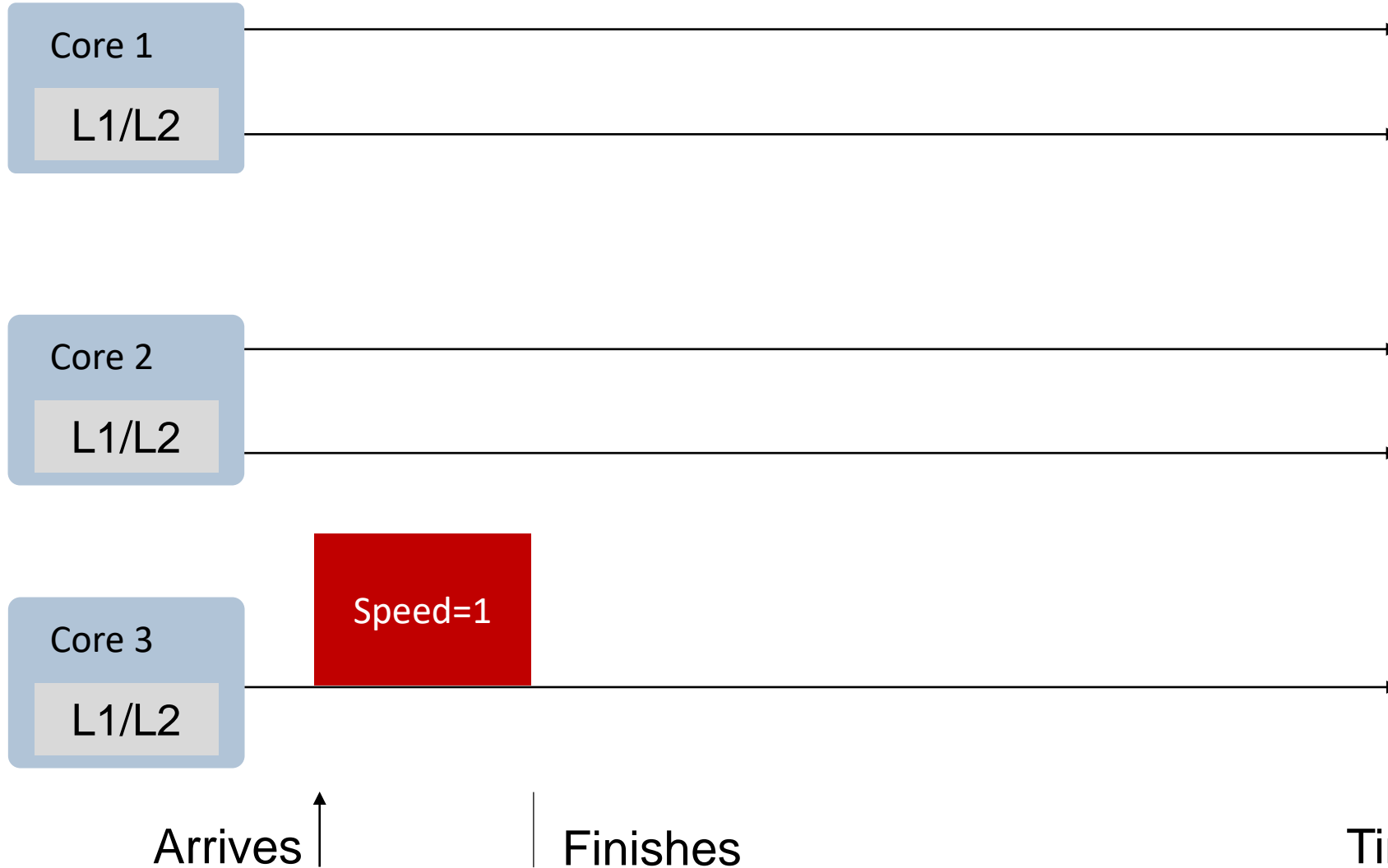
- *All computers are multicores.*
- *Most chip makers do not offer single core.*
- *Most multicores have shared memory.*



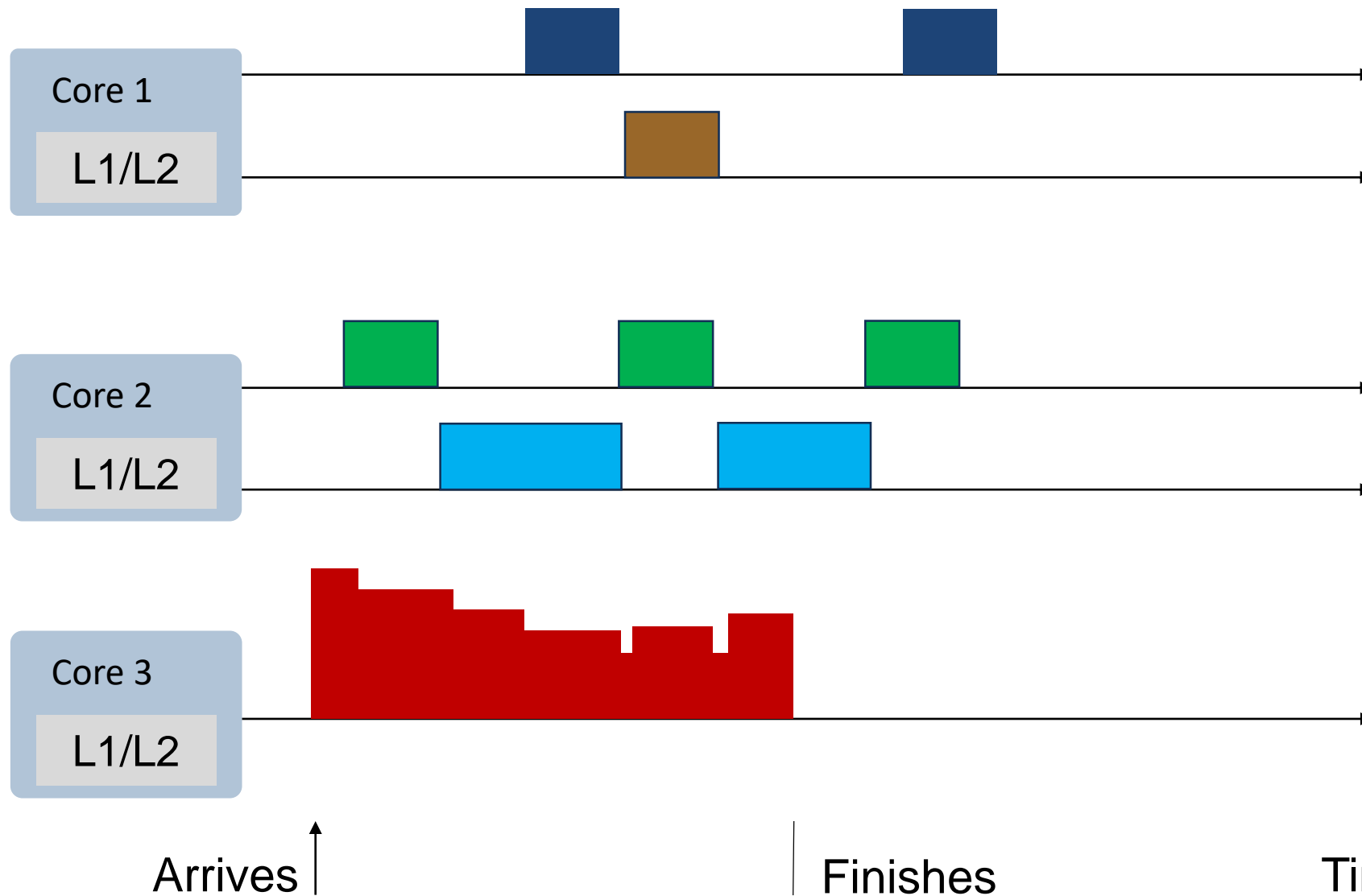
Problem: For each process, compute an upper bound on its response time.



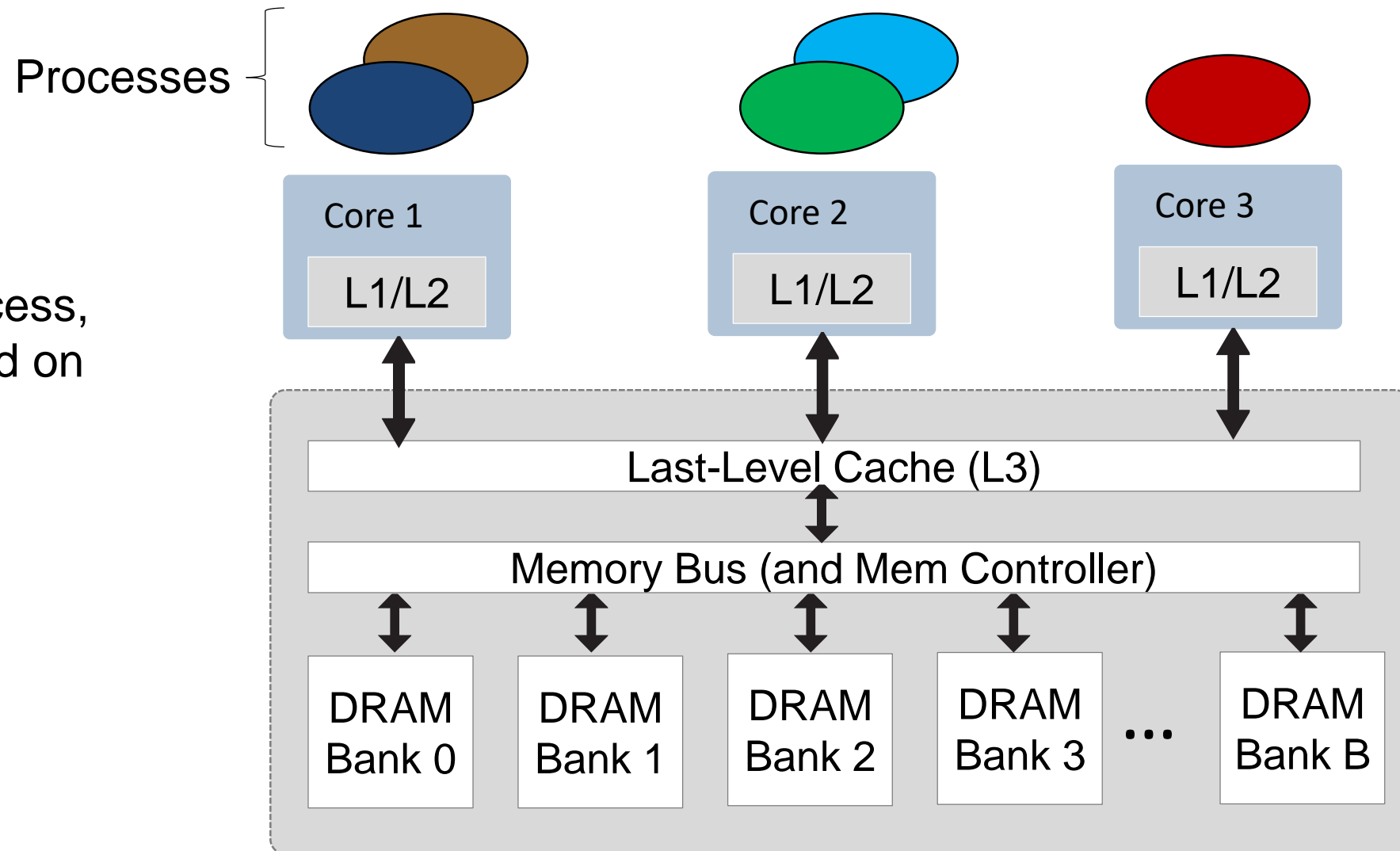
How Co-Runners Impact Speed of Execution



How Co-Runners Impact Speed of Execution



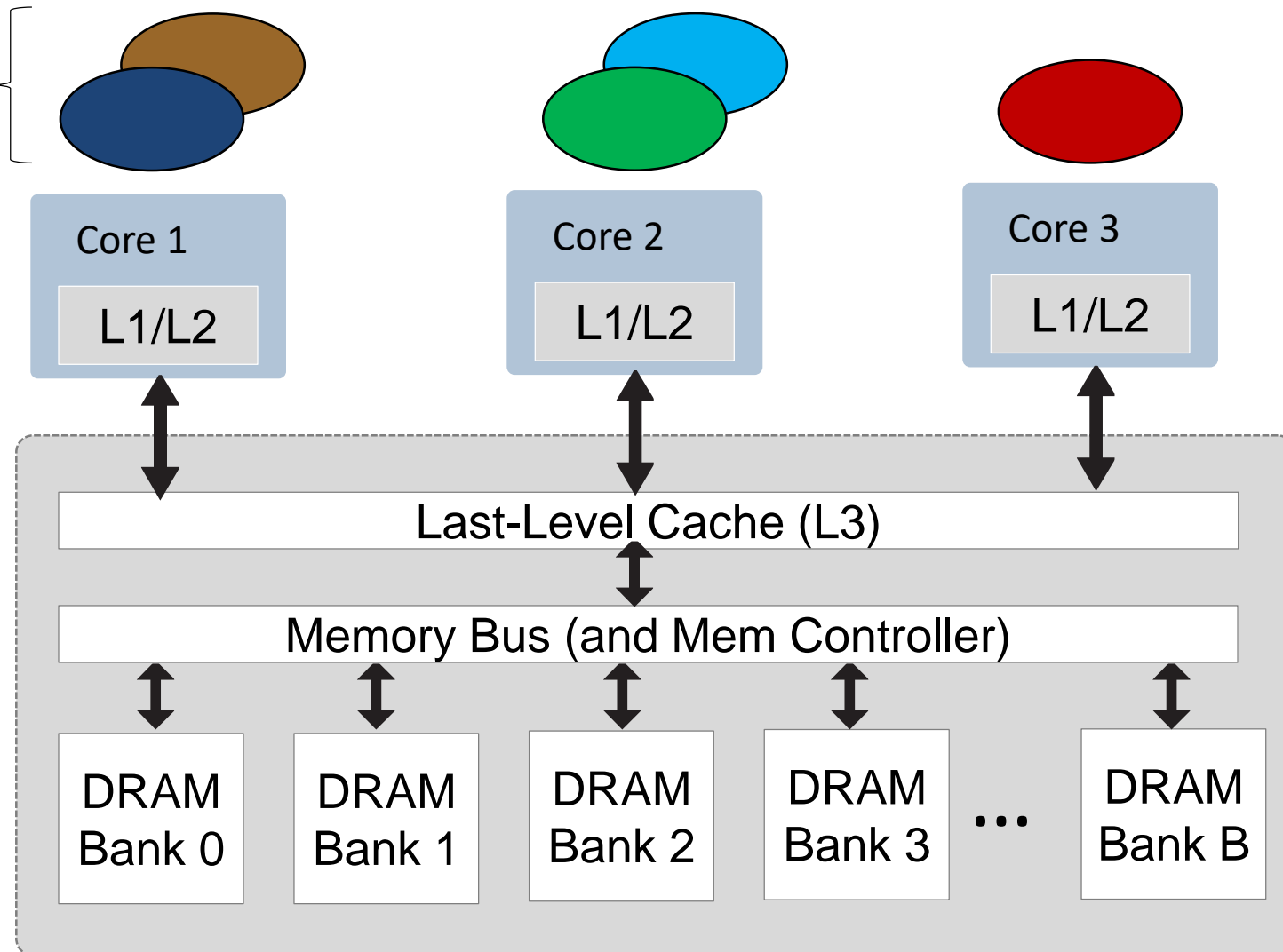
Problem: For each process, compute an upper bound on its response time.



Issues

- *Shared hardware resources impact timing.*

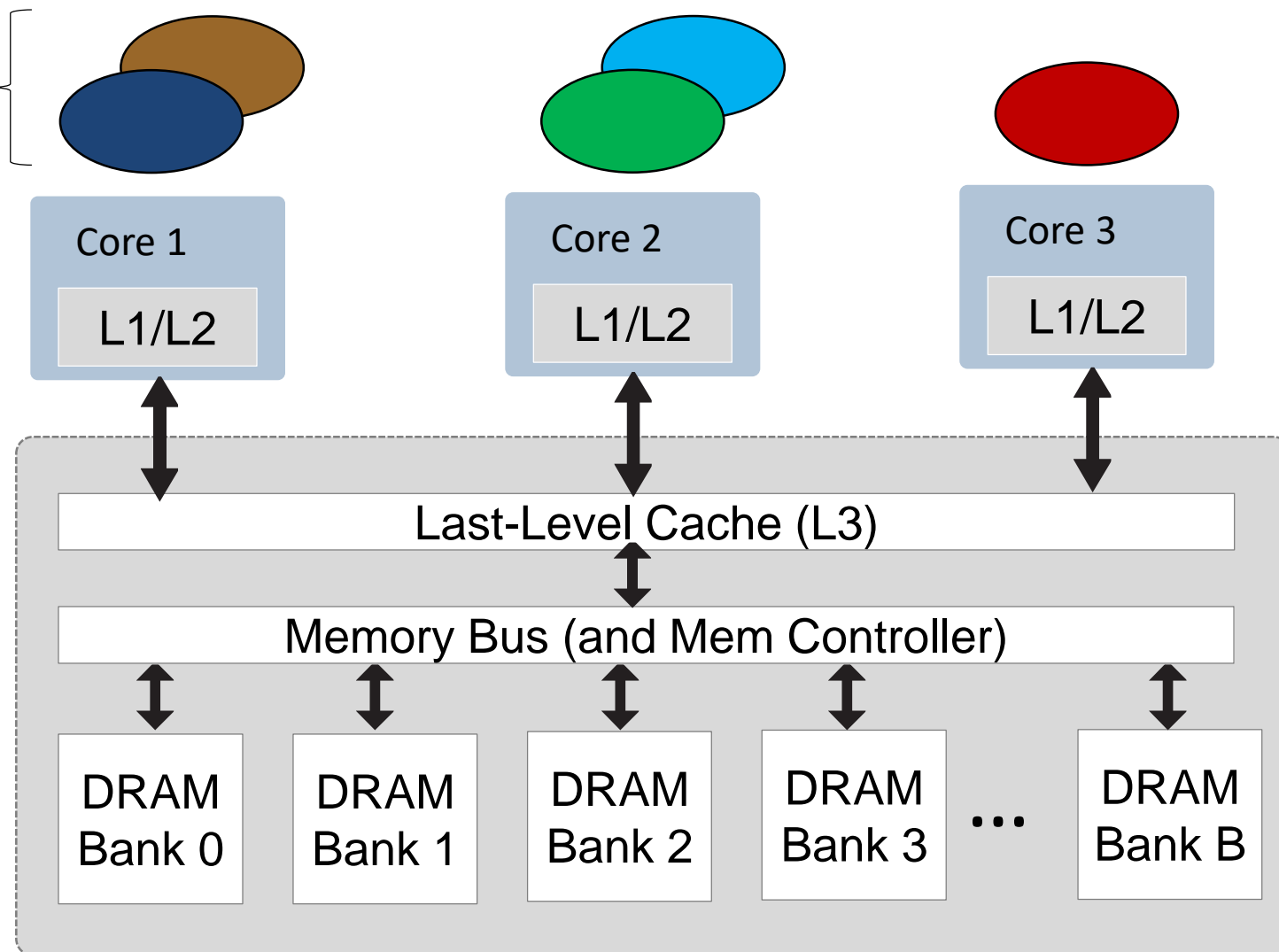
Processes



Issues

- *Shared hardware resources impact timing.*
- *103 times slowdown has been observed.**

Processes

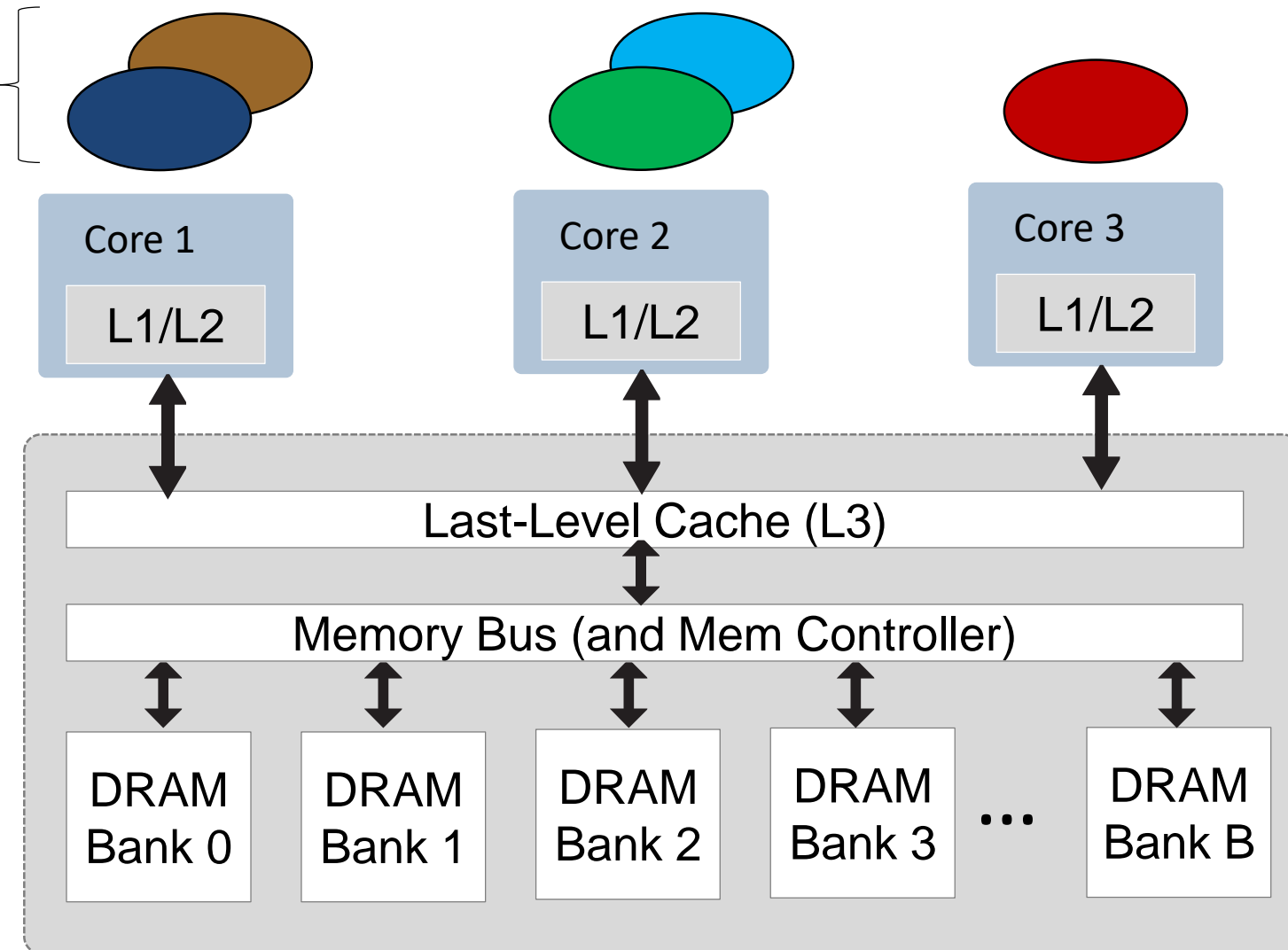


*H. Yun and P. K. Valsan, "Evaluating the Isolation Effect of Cache Partitioning on COTS Multicore Platforms," OSPERT, 2015.

Issues

- *Shared hardware resources impact timing.*
- *103 times slowdown has been observed [Yun15].*
- *Current methods cannot deal with undocumented resources.*

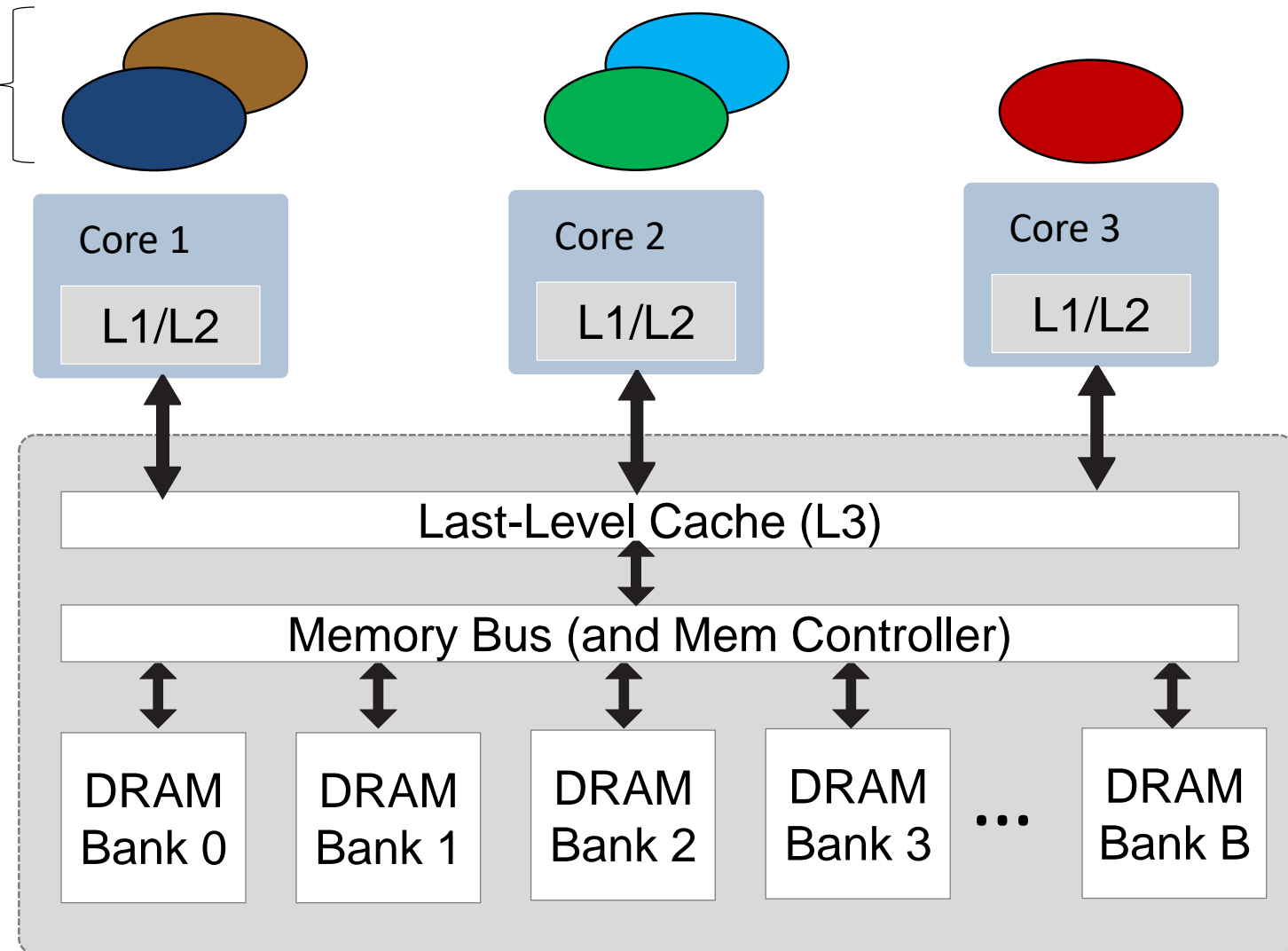
Processes



Issues

- *Shared hardware resources impact timing.*
- *103 times slowdown has been observed [Yun15].*
- *Current methods cannot deal with undocumented resources.*
- *Even for the case that resources are documented, current methods can only analyze/manage a small set of them.*

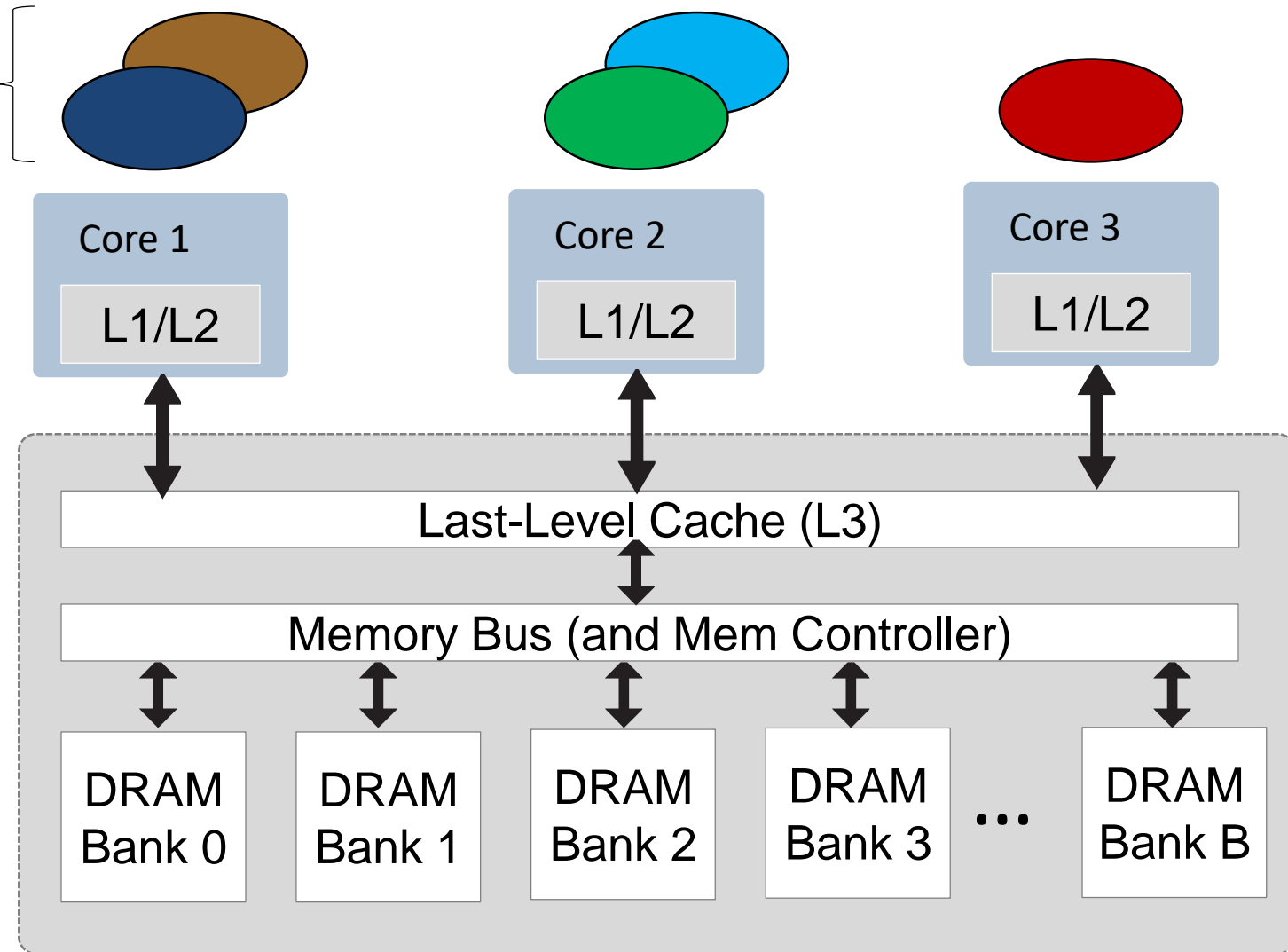
Processes



Issues

- *Shared hardware resources impact timing.*
- *103 times slowdown has been observed [Yun15].*
- *Current methods cannot deal with undocumented resources.*
- *Even when resources are documented, current methods can only analyze/manage a small set of them.*
- *The problem is getting worse:*
 - *Slowdown increasing*
 - *More undocumented h/w*

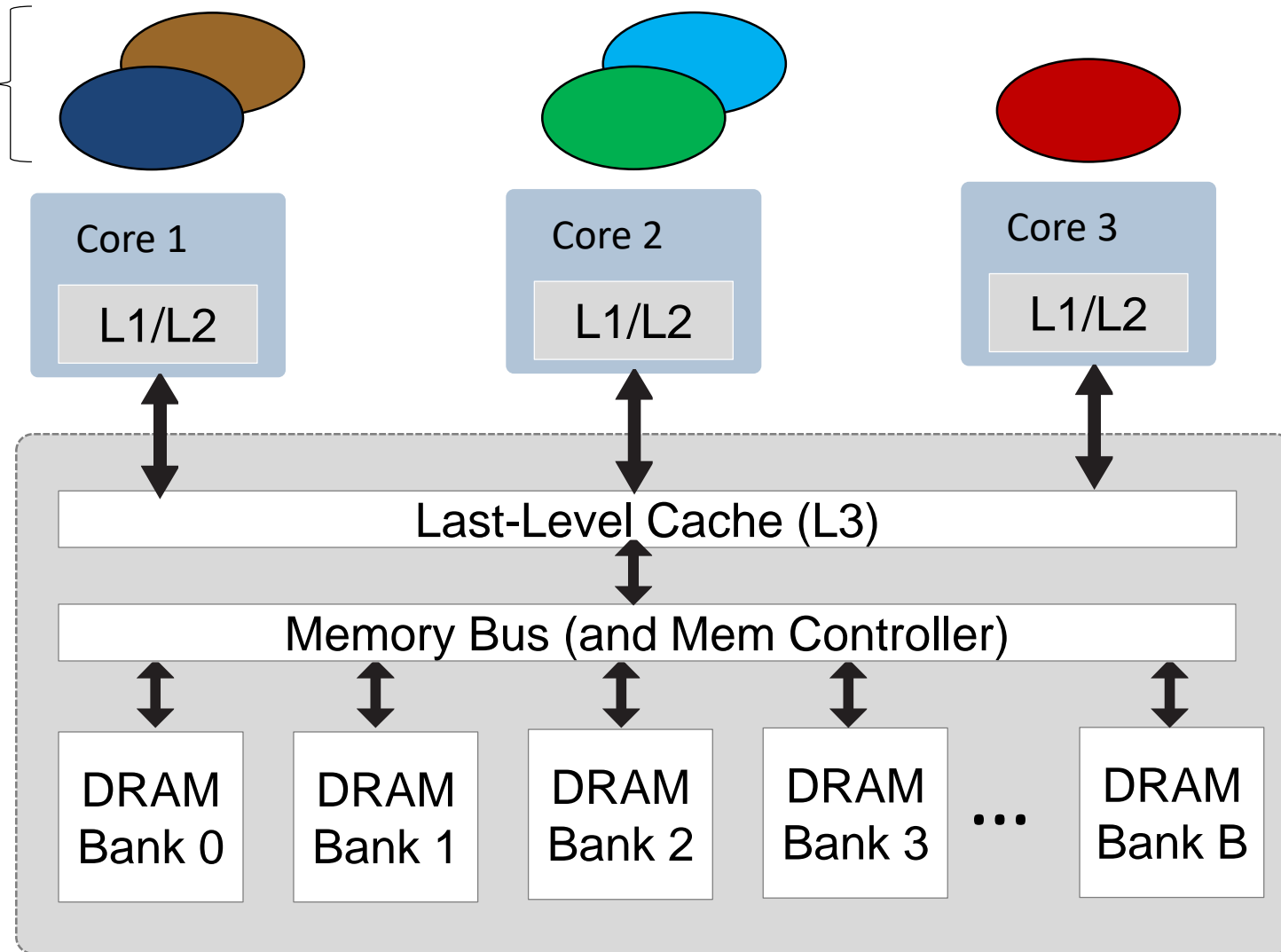
Processes



Issues

- *Shared hardware resources impact timing.*
- *103 times slowdown has been observed [Yun15].*
- *Current methods cannot deal with undocumented resources.*
- *Even when resources are documented, current methods can only analyze/manage a small set of them.*
- *The problem is getting worse:*
 - *Slowdown increasing*
 - *More undocumented h/w*

Processes



We need a new method to compute response times of

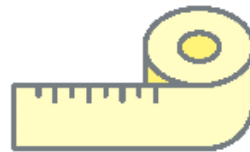
Project Objective

Many real-time systems within the DoD have all processor cores except one disabled in order to be confident about timing.

Therefore, the objective of this project is to develop a solution to overcome this obstacle.



New verification procedure



Method to obtain abstractions



Configuration

A Look at Our Analysis Tool - 1

Schedulability analysis of tasks with co-runner dependent execution times

This program implements the schedulability test in B. Andersson et al., "Schedulability Analysis of Tasks with Co-Runner-Dependent Execution Times," ACM TECS, 2018.

Number of tasks Number of processors

	Minimum inter-arrival time	Deadline	Number of segments	Priority	Processor	Execution requirement	Default speed	Co-runner specification
Task 1	<input type="text" value="1.5"/>	<input type="text" value="1.5"/>	<input type="text" value="1"/>	<input type="text" value="3"/>	<input type="text" value="1"/>			
Segment 1						<input type="text" value="0.25"/>	<input type="text" value="0.5"/>	<input type="text" value="[[[[3, 1]], 1.0], [[4, 1]], 0.5], [[5, 1]], 1.0], [[5, 2]], 1.0]"/>
Segment 2						<input type="text"/>	<input type="text"/>	<input type="text"/>
Segment 3						<input type="text"/>	<input type="text"/>	<input type="text"/>
Task 2	<input type="text" value="2.0"/>	<input type="text" value="2.0"/>	<input type="text" value="1"/>	<input type="text" value="2"/>	<input type="text" value="1"/>			
Segment 1						<input type="text" value="0.25"/>	<input type="text" value="0.5"/>	<input type="text" value="[[[[3, 1]], 0.5], [[4, 1]], 1.0], [[5, 1]], 1.0], [[5, 2]], 1.0]"/>
Segment 2						<input type="text"/>	<input type="text"/>	<input type="text"/>
Segment 3						<input type="text"/>	<input type="text"/>	<input type="text"/>
Task 3	<input type="text" value="2.0"/>	<input type="text" value="2.0"/>	<input type="text" value="1"/>	<input type="text" value="3"/>	<input type="text" value="2"/>			
Segment 1						<input type="text" value="0.25"/>	<input type="text" value="0.5"/>	<input type="text" value="[[[[1, 1]], 1.0], [[2, 1]], 0.5]"/>
Segment 2						<input type="text"/>	<input type="text"/>	<input type="text"/>
Segment 3						<input type="text"/>	<input type="text"/>	<input type="text"/>
Task 4	<input type="text" value="2.0"/>	<input type="text" value="2.0"/>	<input type="text" value="1"/>	<input type="text" value="2"/>	<input type="text" value="2"/>			
Segment 1						<input type="text" value="0.25"/>	<input type="text" value="0.5"/>	<input type="text" value="[[[[1, 1]], 0.5], [[2, 1]], 1.0]"/>
Segment 2						<input type="text"/>	<input type="text"/>	<input type="text"/>
Segment 3						<input type="text"/>	<input type="text"/>	<input type="text"/>
Task 5	<input type="text" value="2.25"/>	<input type="text" value="2.25"/>	<input type="text" value="2"/>	<input type="text" value="1"/>	<input type="text" value="2"/>			
Segment 1						<input type="text" value="0.5"/>	<input type="text" value="1.0"/>	<input type="text" value="[[[[1, 1]], 1.0], [[2, 1]], 1.0]"/>
Segment 2						<input type="text" value="0.125"/>	<input type="text" value="0.5"/>	<input type="text" value="[[[[1, 1]], 0.5], [[2, 1]], 1.0]"/>
Segment 3						<input type="text"/>	<input type="text"/>	<input type="text"/>

A Look at Our Analysis Tool - 2

Schedulability analysis of tasks with co-runner dependent execution times

This program implements the schedulability test in B. Andersson et al., "Schedulability Analysis of Tasks with Co-Runner-Dependent Execution Times," ACM TECS, 2018.

Number of tasks Number of processors

	Minimum inter-arrival time	Deadline	Number of segments	Priority	Processor	Execution requirement	Default speed	Co-runner specification
Task 1	<input type="text" value="1.5"/>	<input type="text" value="1.5"/>	<input type="text" value="1"/>	<input type="text" value="3"/>	<input type="text" value="1"/>			
Segment 1						<input type="text" value="0.25"/>	<input type="text" value="0.5"/>	<input type="text" value="[[[[3, 1]], 1.0], [[4, 1]], 0.5], [[5, 1]], 1.0], [[5, 2]], 1.0]"/>
Segment 2								
Segment 3								
Task 2	<input type="text" value="2.0"/>	<input type="text" value="2.0"/>	<input type="text" value="1"/>	<input type="text" value="2"/>	<input type="text" value="1"/>			
Segment 1								<input type="text" value="1]], 1.0], [[5, 1]], 1.0], [[5, 2]], 1.0]"/>
Segment 2								
Segment 3								
Task 3	<input type="text" value="2.0"/>	<input type="text" value="2.0"/>	<input type="text" value="1"/>	<input type="text" value="3"/>	<input type="text" value="1"/>			
Segment 1								<input type="text" value="1]], 0.5]"/>
Segment 2								
Segment 3								
Task 4	<input type="text" value="2.0"/>	<input type="text" value="2.0"/>	<input type="text" value="1"/>	<input type="text" value="2"/>	<input type="text" value="1"/>			
Segment 1						<input type="text" value="0.25"/>	<input type="text" value="0.5"/>	<input type="text" value="[[[[4, 1]], 0.5], [[5, 1]], 1.0]"/>
Segment 2								
Segment 3								
Task 5	<input type="text" value="2.25"/>	<input type="text" value="2.25"/>	<input type="text" value="2"/>	<input type="text" value="1"/>	<input type="text" value="2"/>			
Segment 1						<input type="text" value="0.5"/>	<input type="text" value="1.0"/>	<input type="text" value="[[[[1, 1]], 1.0], [[2, 1]], 1.0]"/>
Segment 2						<input type="text" value="0.125"/>	<input type="text" value="0.5"/>	<input type="text" value="[[[[1, 1]], 0.5], [[2, 1]], 1.0]"/>
Segment 3								

Taskset is schedulable

Upper bounds on the response times of task are as follows:

For task 1: 0.5

For task 2: 1.0

For task 3: 0.5

For task 4: 1.0

For task 5: 1.75