Carnegie Mellon University Software Engineering Institute

Research Review 2021

Using All Processor Cores While Being Confident about Timing

November 2021

Bjorn Andersson

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Carnegie Mellon University Software Engineering Institute

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.

DM21-0810

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

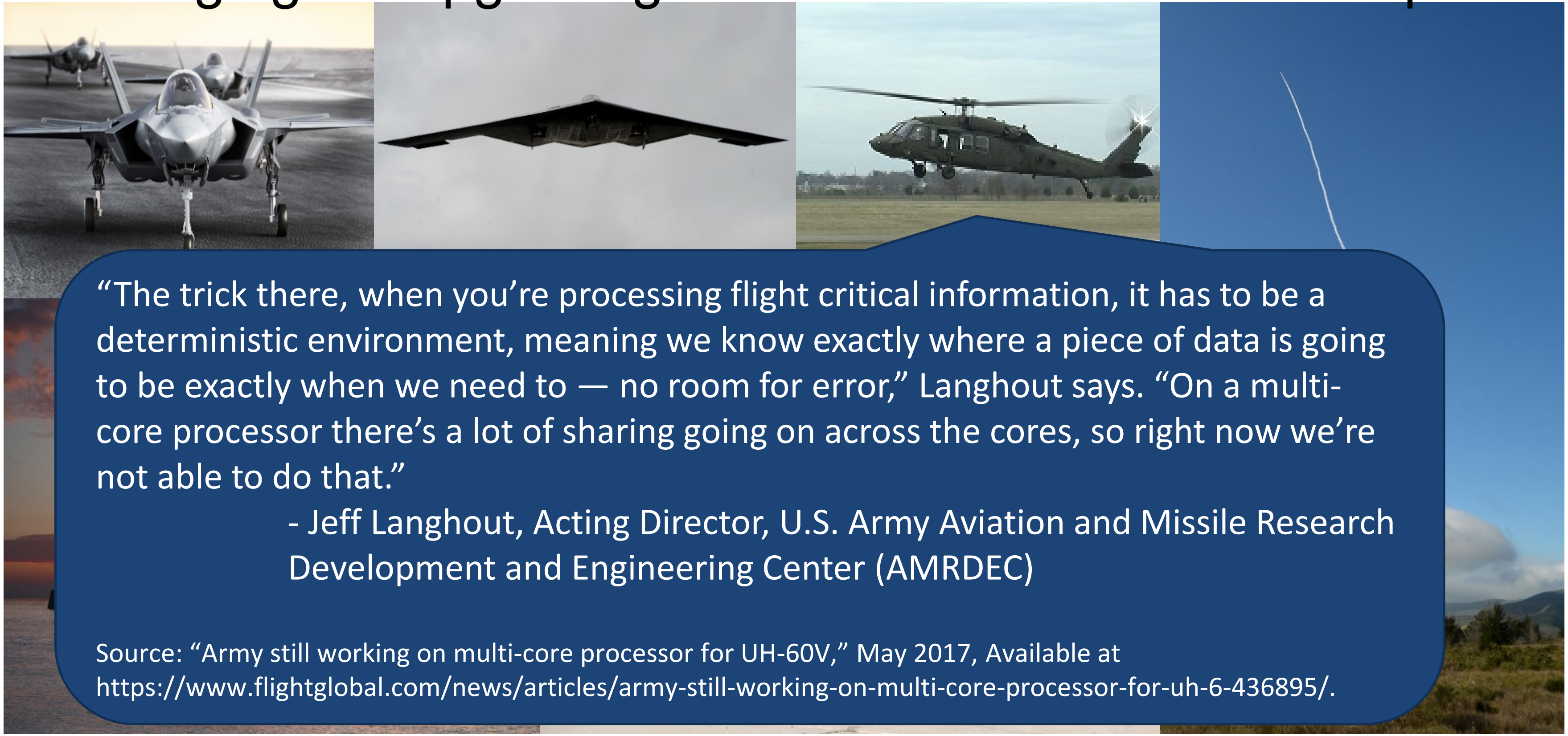





Satisfying Real-Time Requirements Is Challenging for Upgrading the Blackhawk UH-60 Helicopter

Carnegie Mellon University Software Engineering Institute

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

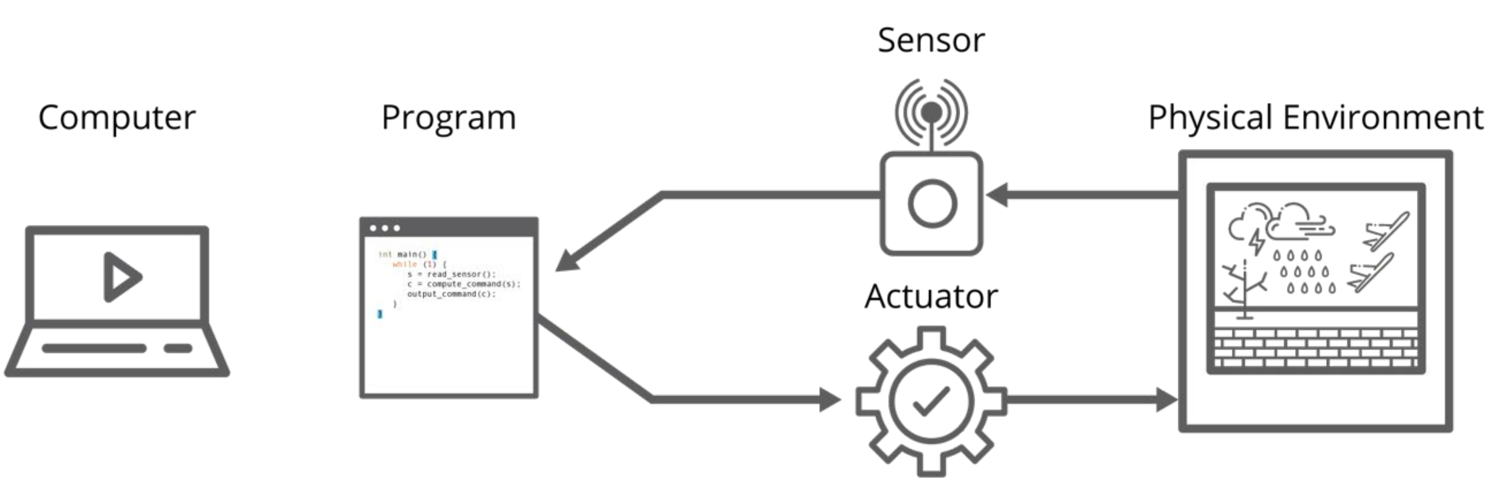


"The trick there, when you're processing flight critical information, it has to be a deterministic environment, meaning we know exactly where a piece of data is going to be exactly when we need to — no room for error," Langhout says. "On a multi-core processor there's a lot of sharing going on across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

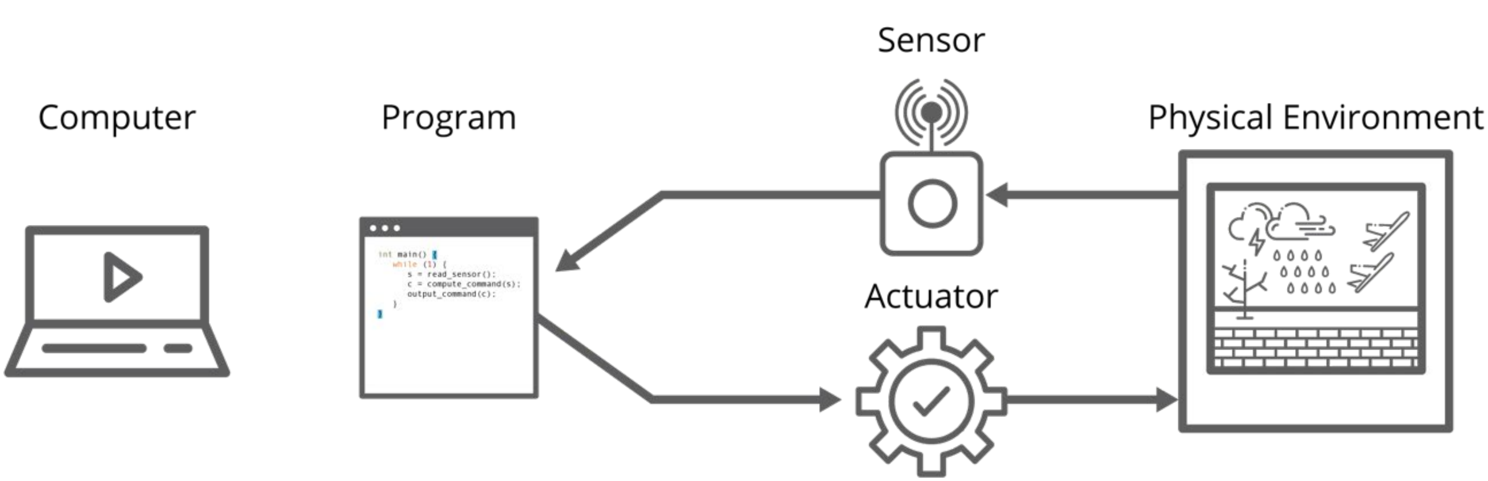
- Jeff Langhout, Acting Director, U.S. Army Aviation and Missile Research Development and Engineering Center (AMRDEC)

Source: "Army still working on multi-core processor for UH-60V," May 2017, Available at 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

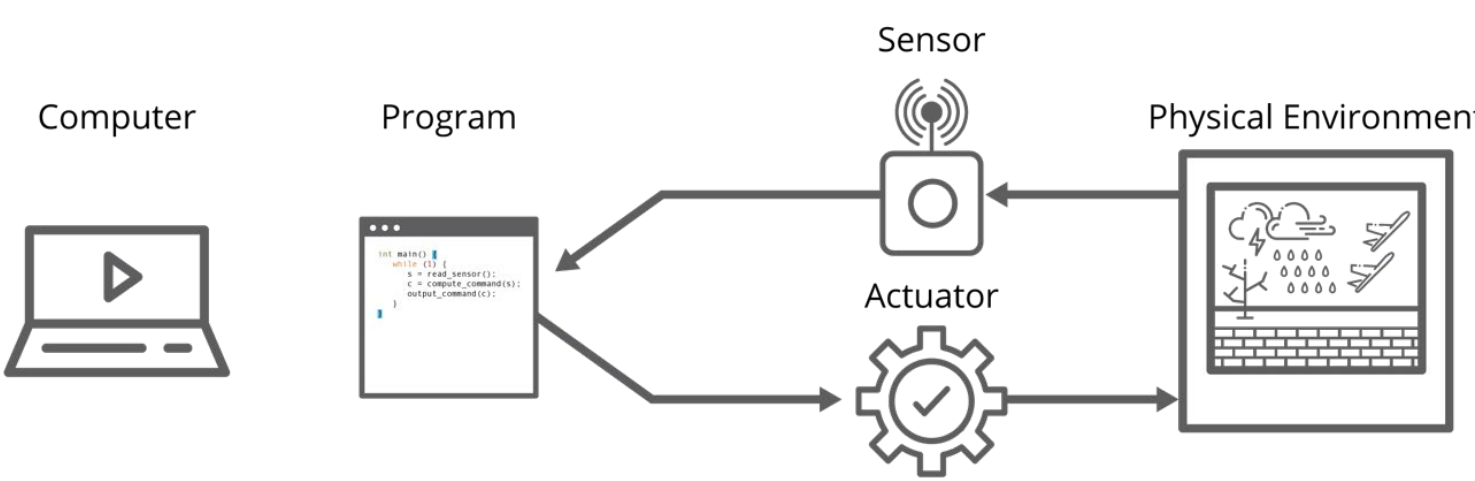


Time

Read Sensor

Actuate Command

Commonality of DoD Systems



Time

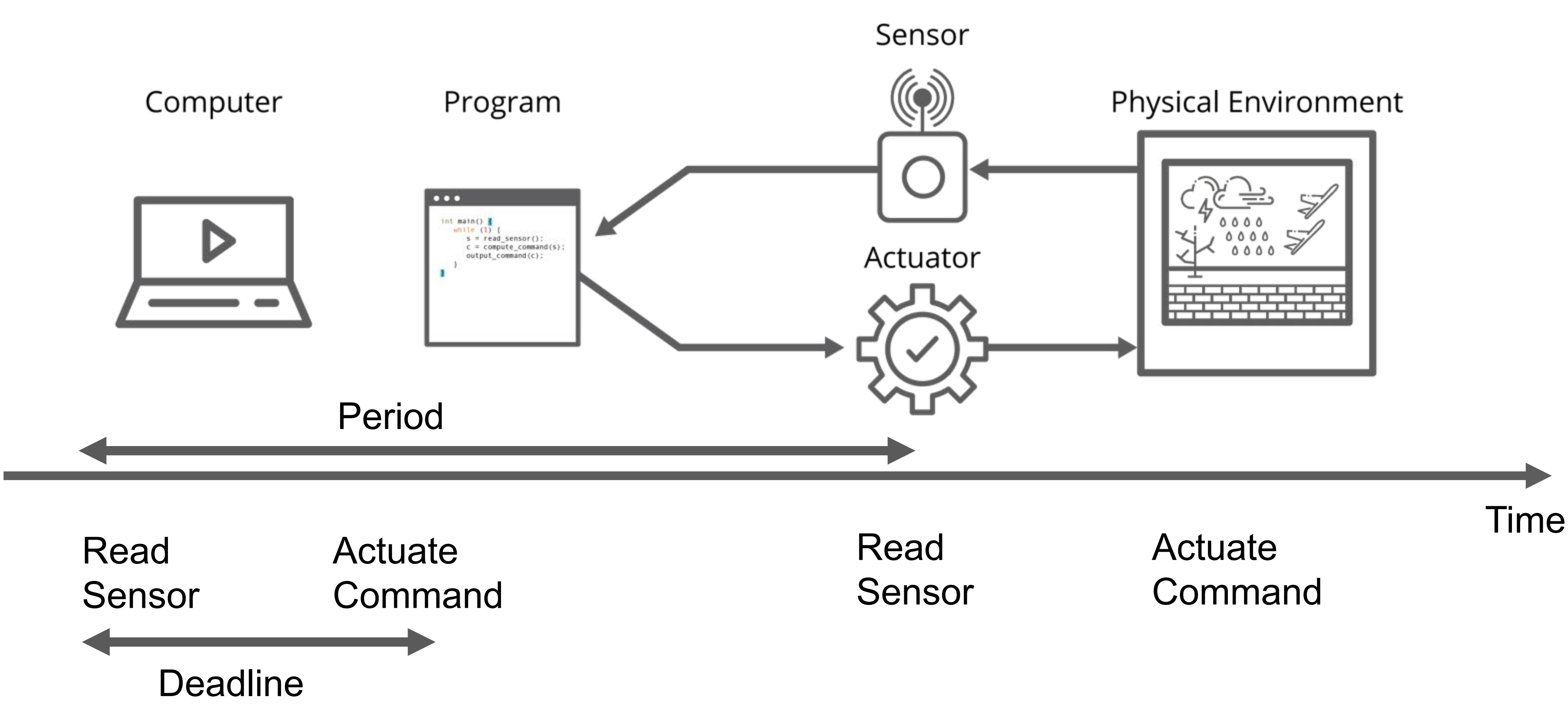
Read Sensor

Actuate Command

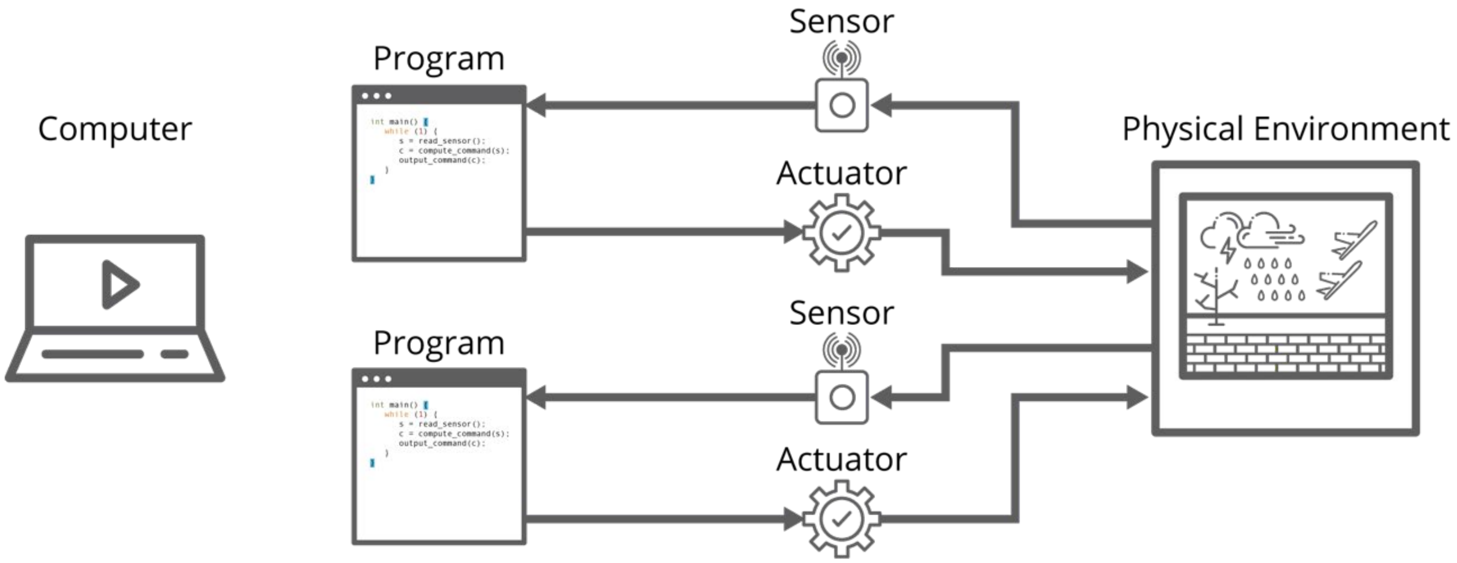
Read Sensor

Actuate Command

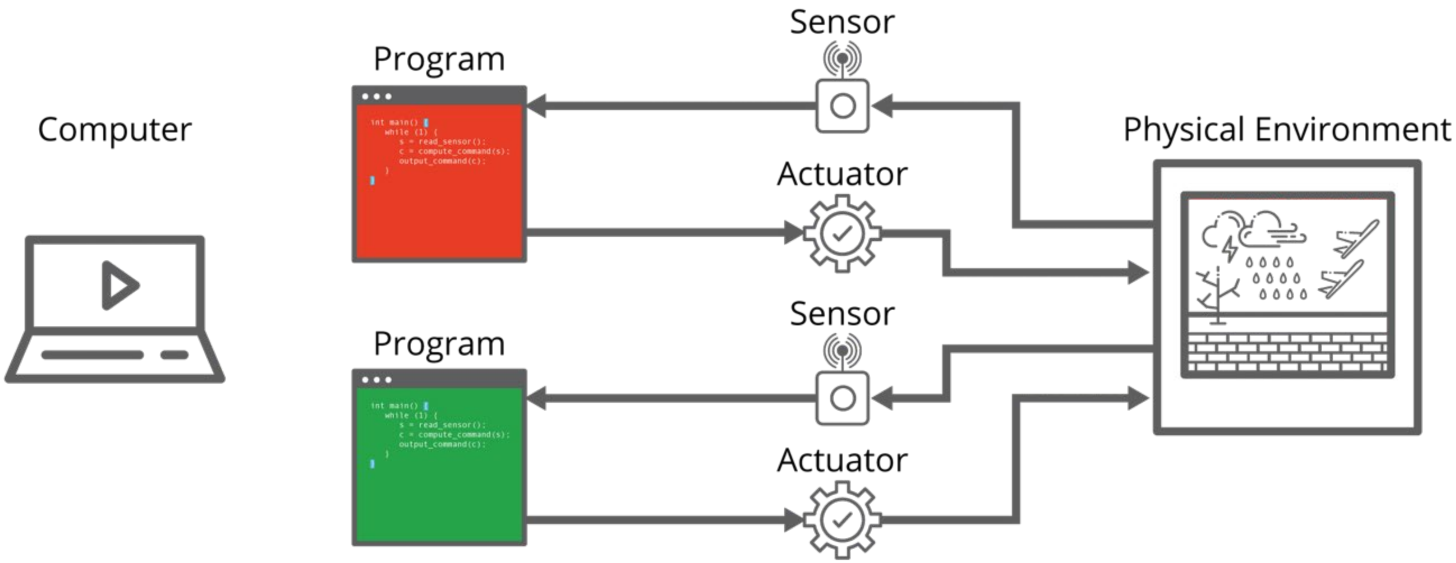
Commonality of DoD Systems



Commonality of DoD Systems



Commonality of DoD Systems



Time

Read Sensor | Actuate Command | Read Sensor | Actuate Command

Read Sensor | Actuate Command | Read Sensor | Actuate Command

Why is Satisfying Real-Time Requirements Challenging?

What Causes Delay of Software?

What Causes Delay of Software?

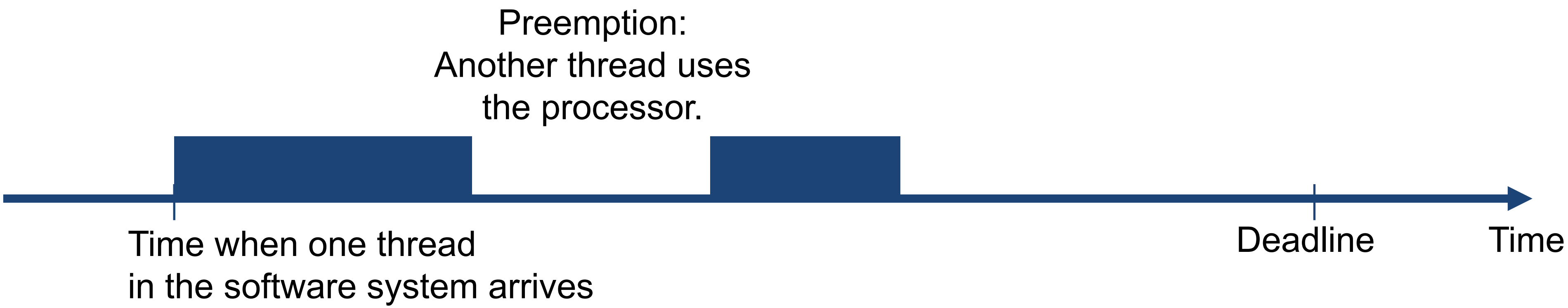
What Causes Delay of Software?

What Causes Delay of Software?

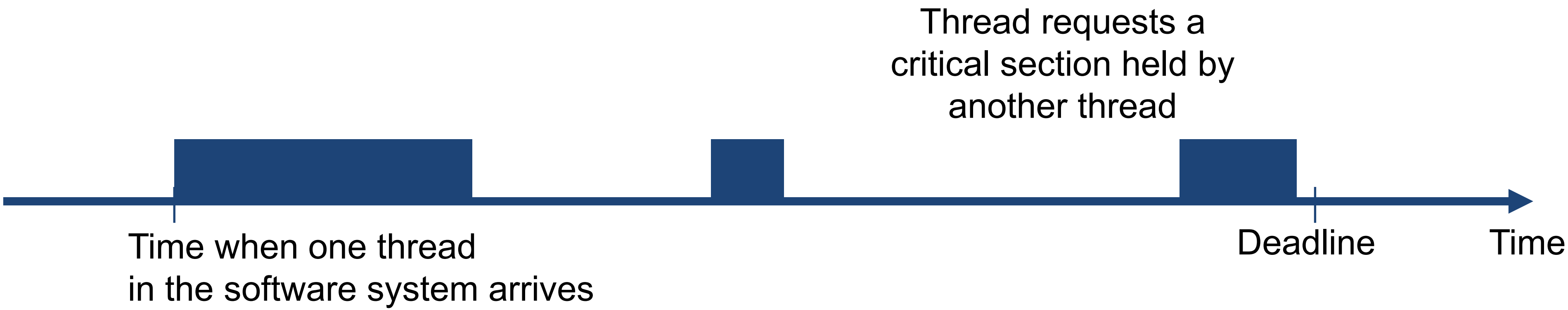
What Causes Delay of Software?

Carnegie Mellon University Software Engineering Institute

What Causes Delay of Software?



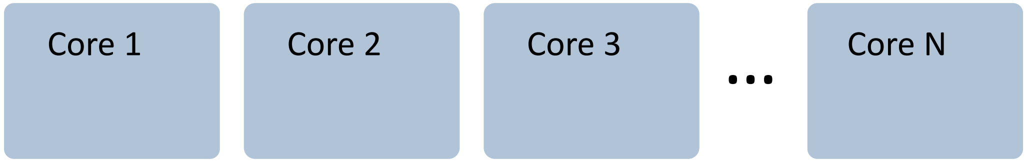
What Causes Delay of Software?



Real-Time Requirements of Software Executing on a Multicore Processor

Hardware Trends

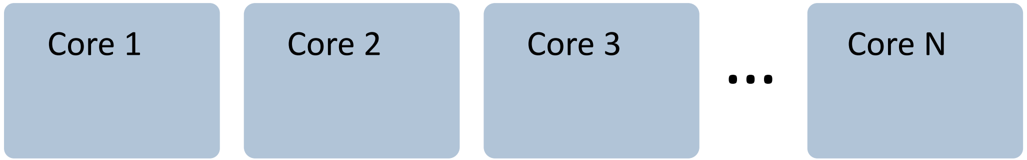
- *All computers are multicores.*



Real-Time Requirements of Software Executing on a Multicore Processor

Hardware Trends

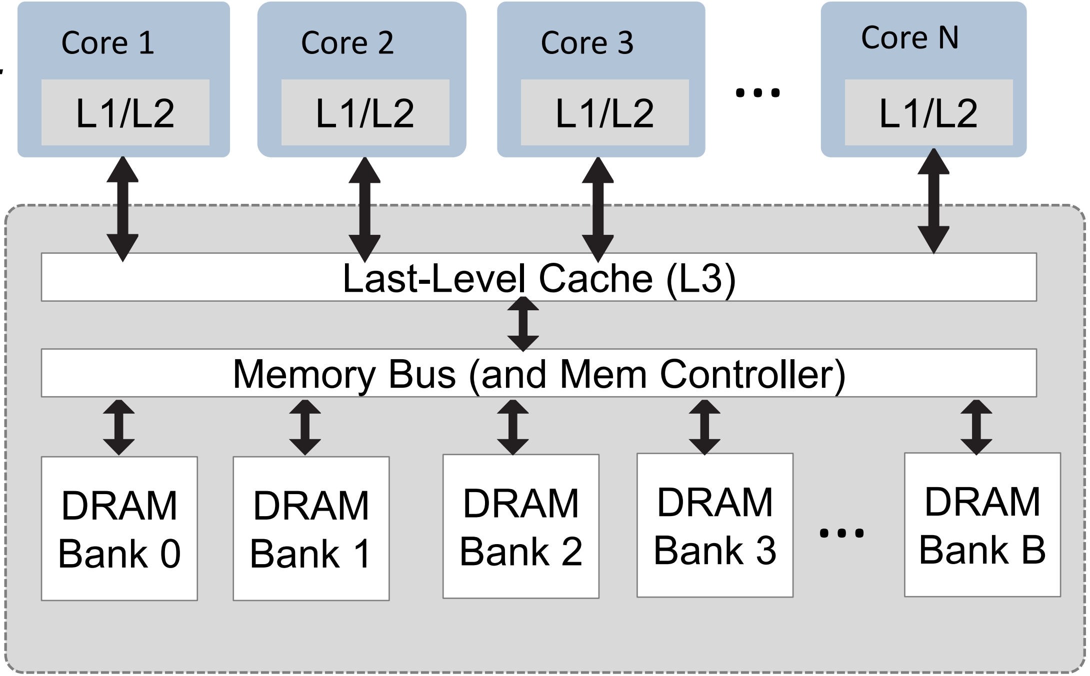
- *All computers are multicores.*
- *Most chip makers do not offer single core.*



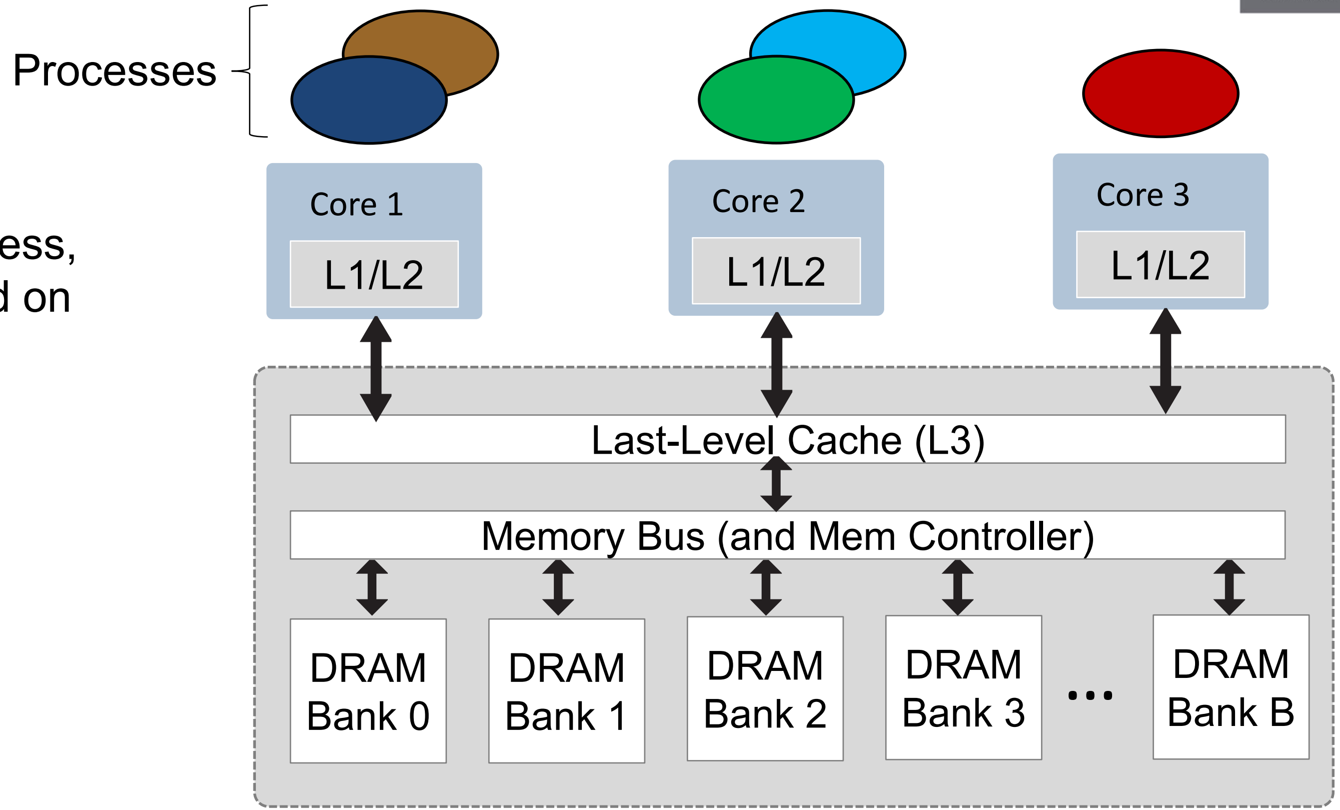
Real-Time Requirements of Software Executing on a Multicore Processor

Hardware Trends

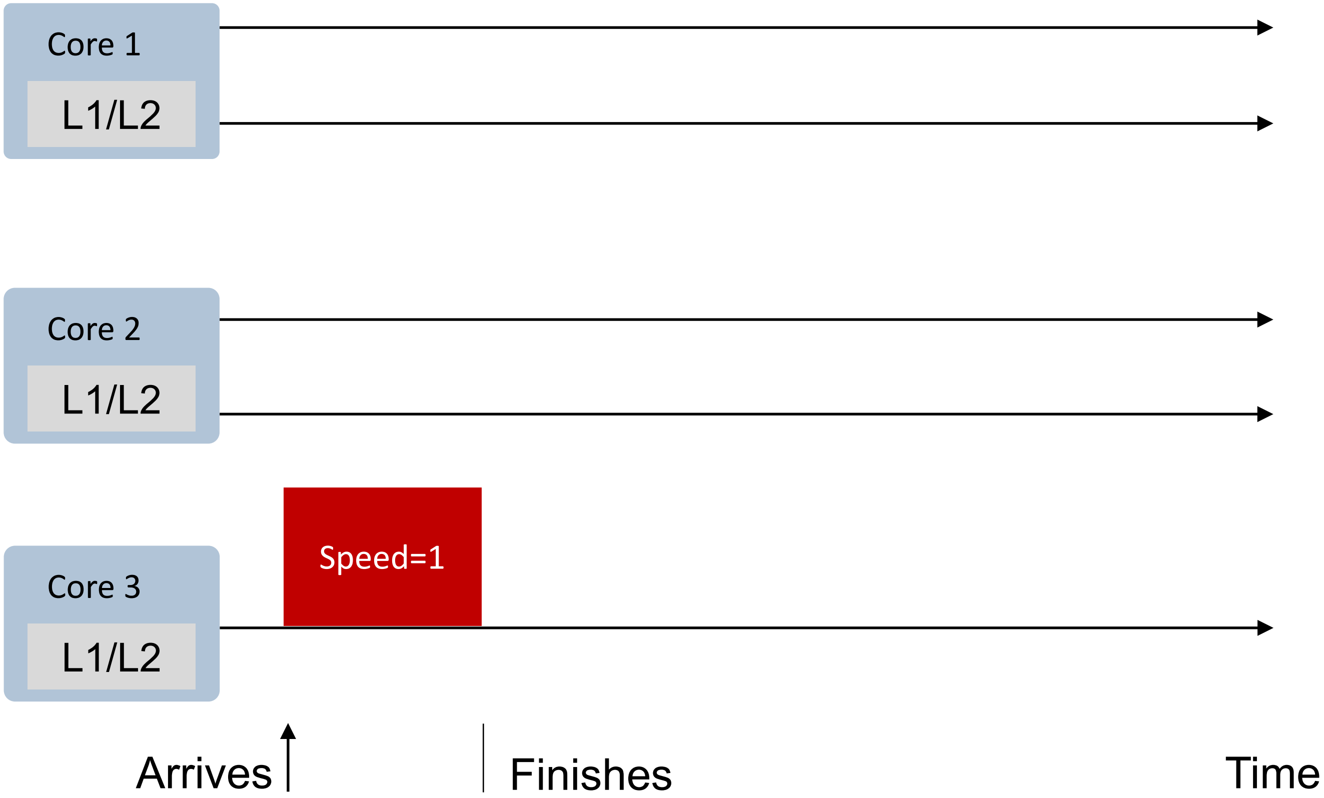
- *All computers are multicores.*
- *Most chip makers do not offer single core.*
- *Most multicores have shared memory.*



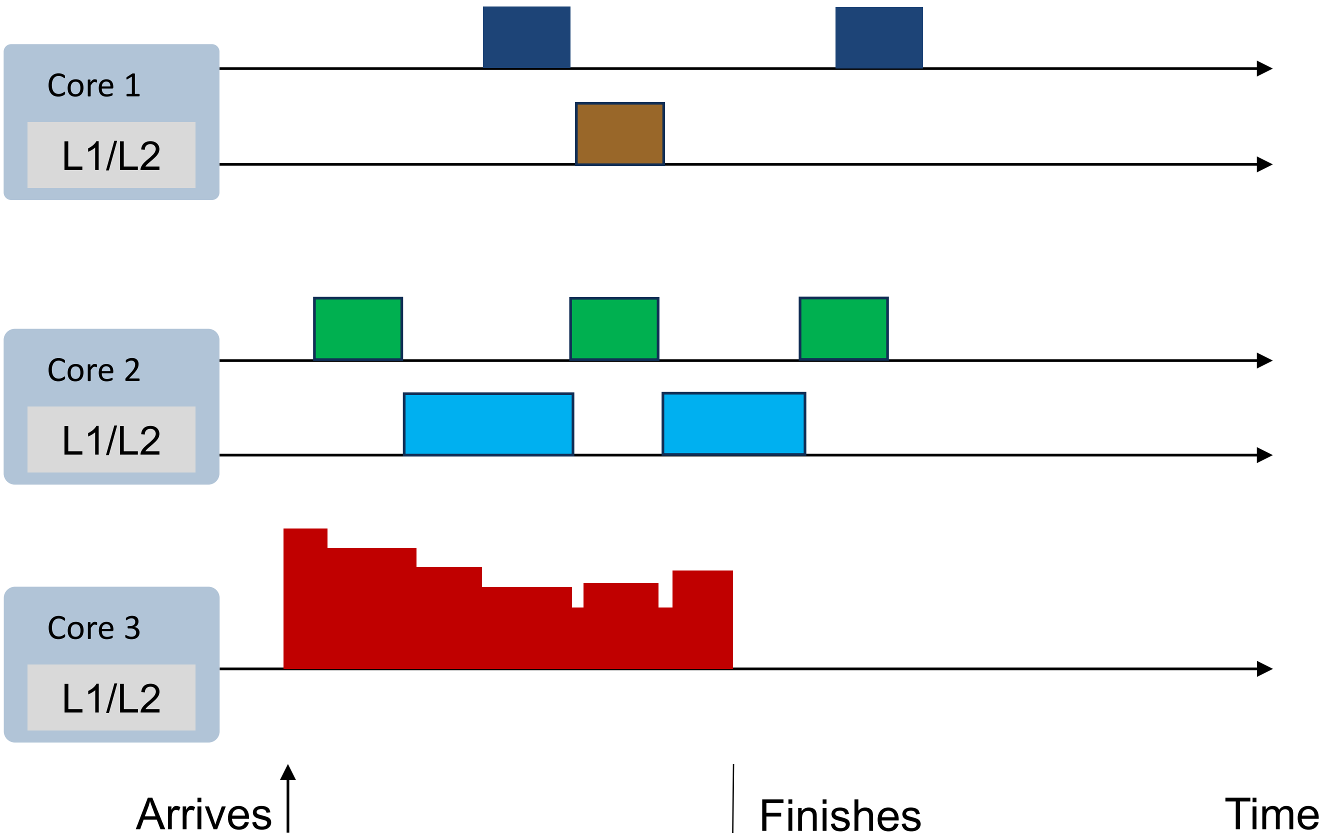
Problem: For each process, compute an upper bound on its response time.



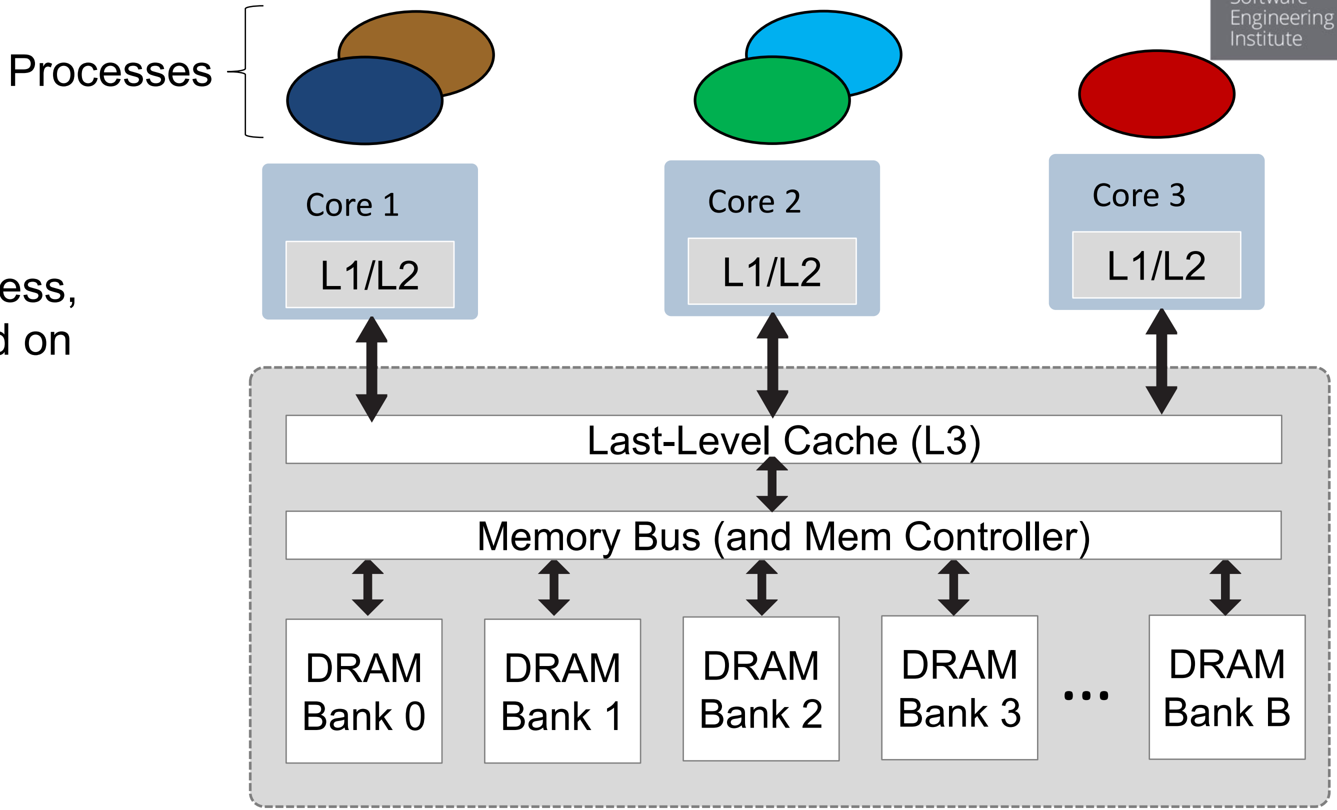
How Co-Runners Impact Speed of Execution



How Co-Runners Impact Speed of Execution



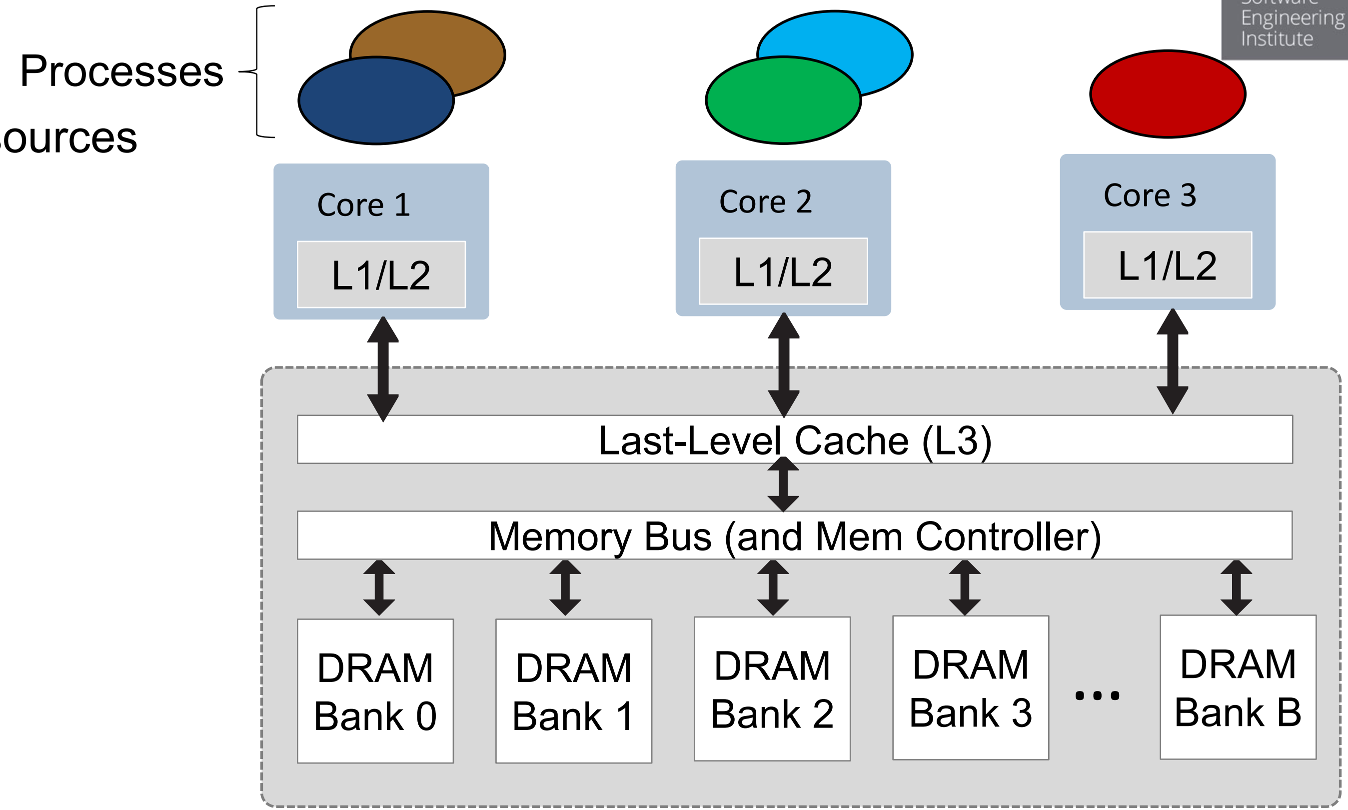
Problem: For each process, compute an upper bound on its response time.



Carnegie Mellon University Software Engineering Institute

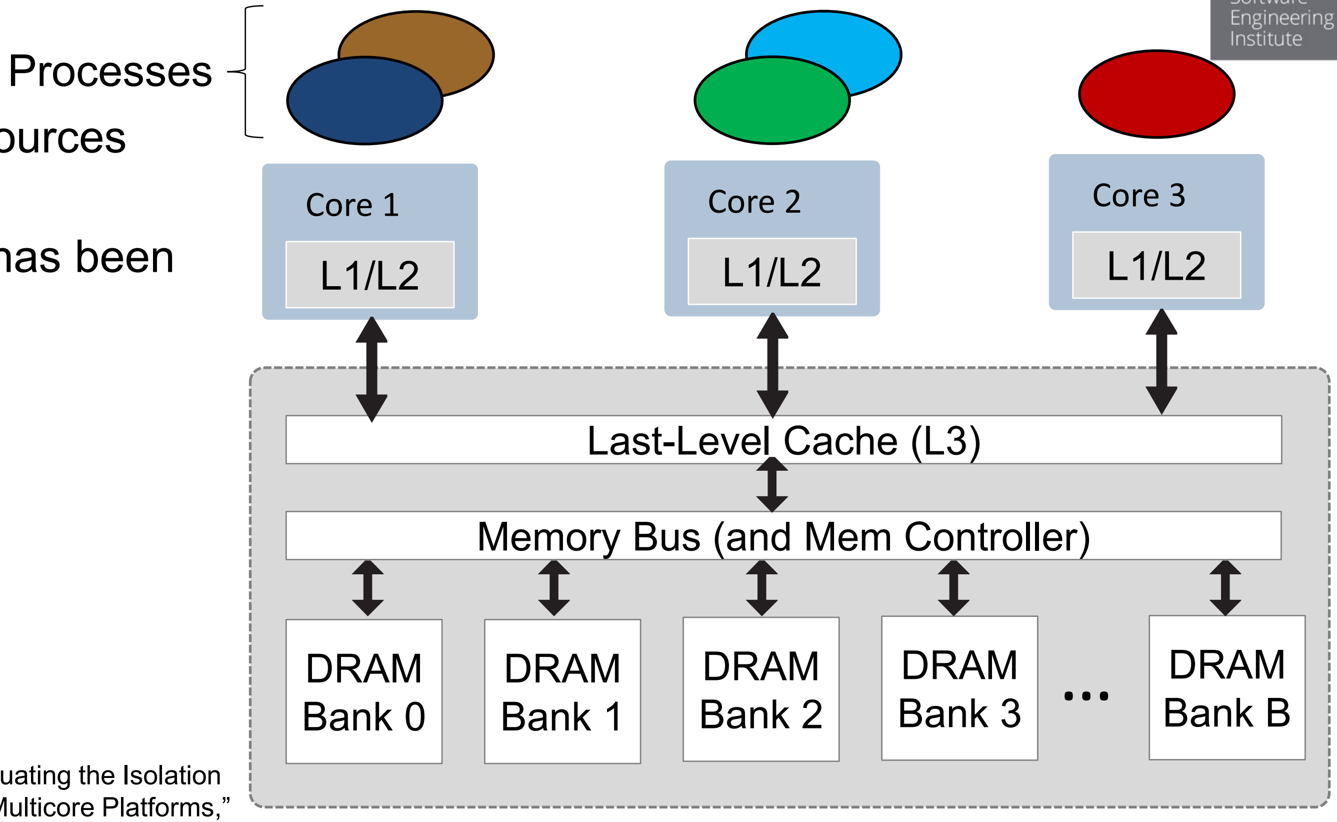
Issues

- Shared hardware resources impact timing.



Issues

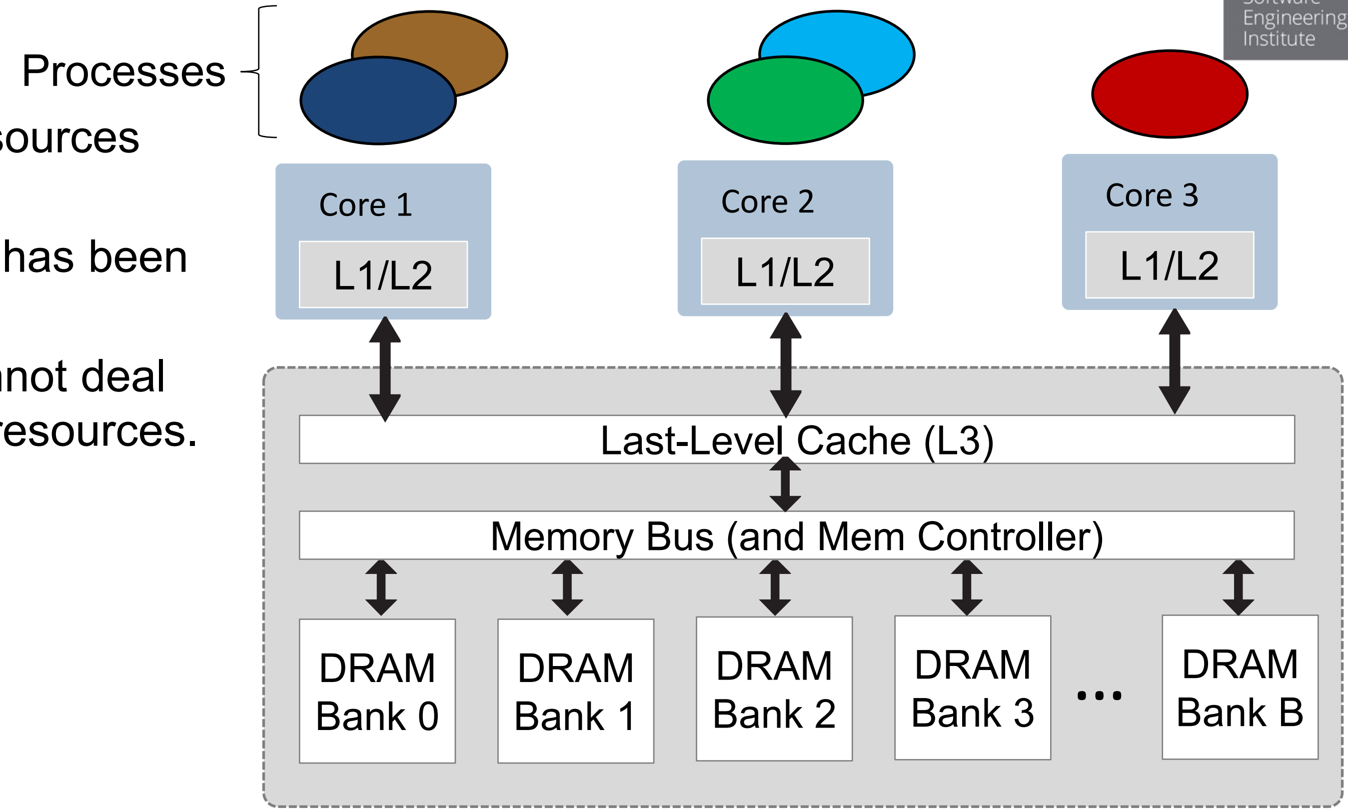
- Shared hardware resources impact timing.
- 103 times slowdown has been observed [Yun15].



[Yun15] H. Yun and P. K. Valsan, "Evaluating the Isolation Effect of Cache Partitioning on COTS Multicore Platforms," OSPERT, 2015.

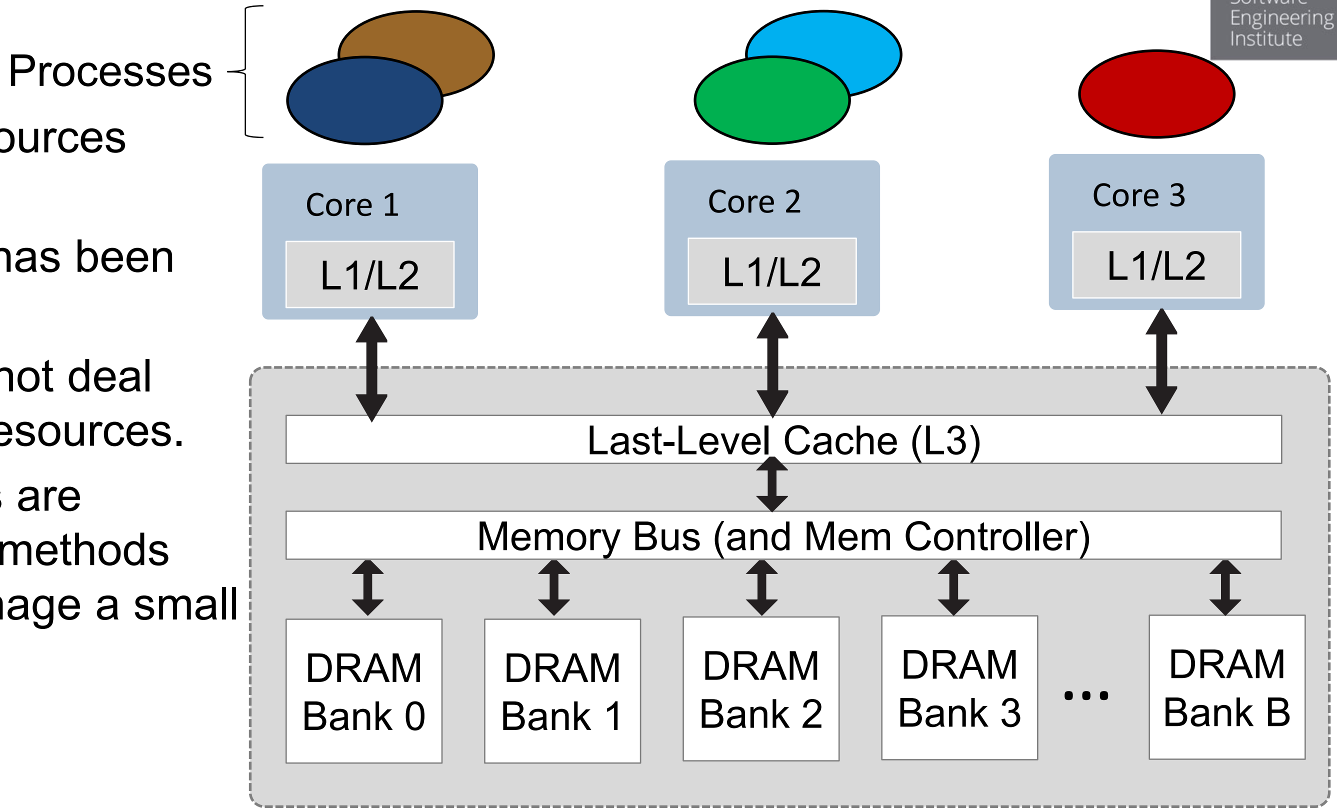
Issues

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



Issues

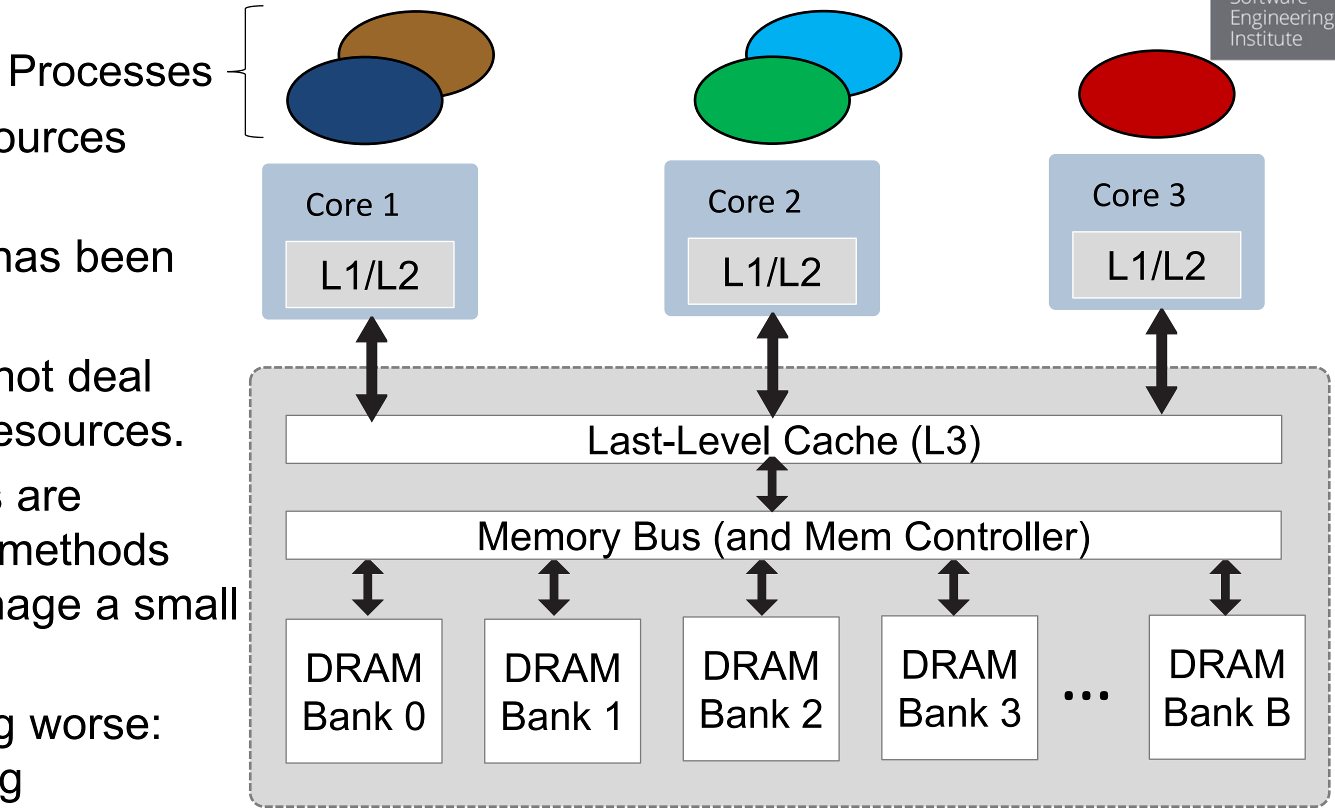
- Shared hardware resources impact timing.
- 103 times slowdown has been observed [Yun15].
- Current methods cannot deal with undocumented resources.
- Even when resources are documented, current methods can only analyze/manage a small set of them.



Carnegie Mellon University Software Engineering Institute

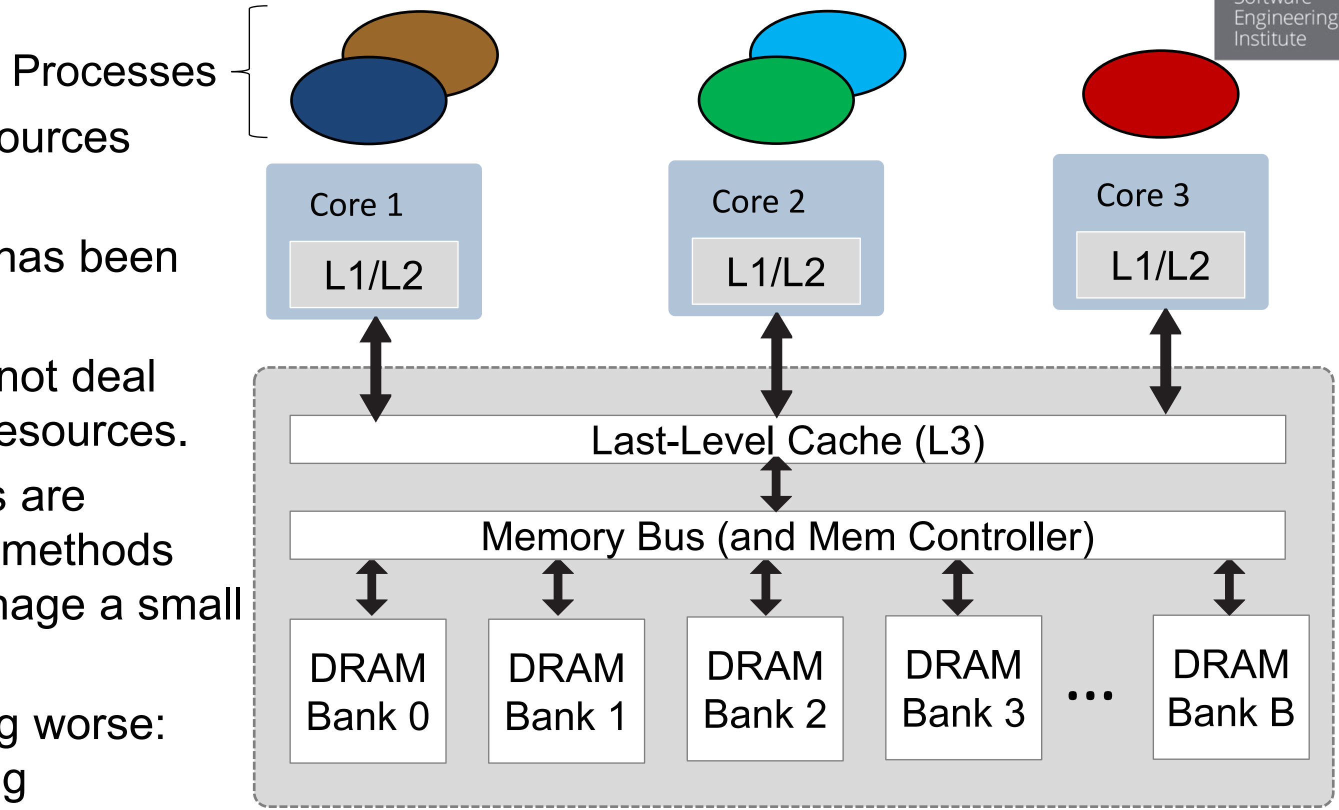
Issues

- 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



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



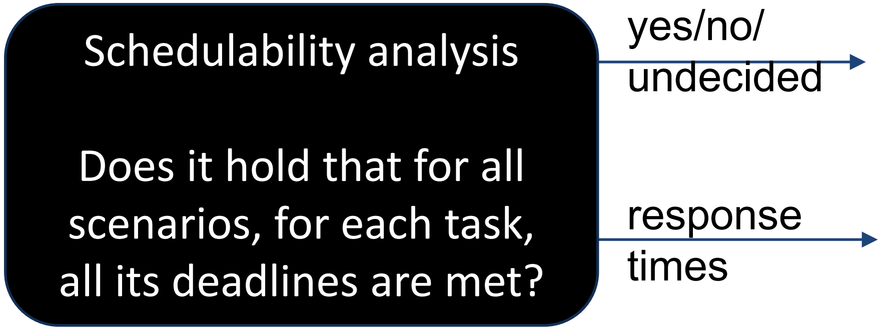
We need a new method to compute response times of processes.

Overview of Our Solution

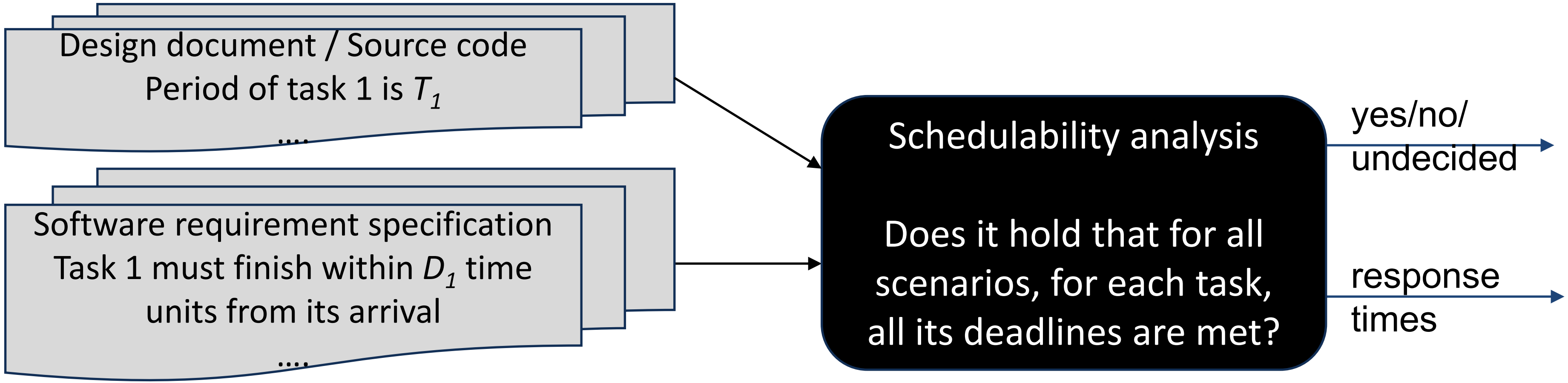
Does it hold that for all scenarios, for each task, all its deadlines are met?

yes/no/ undecided

Overview of Our Solution

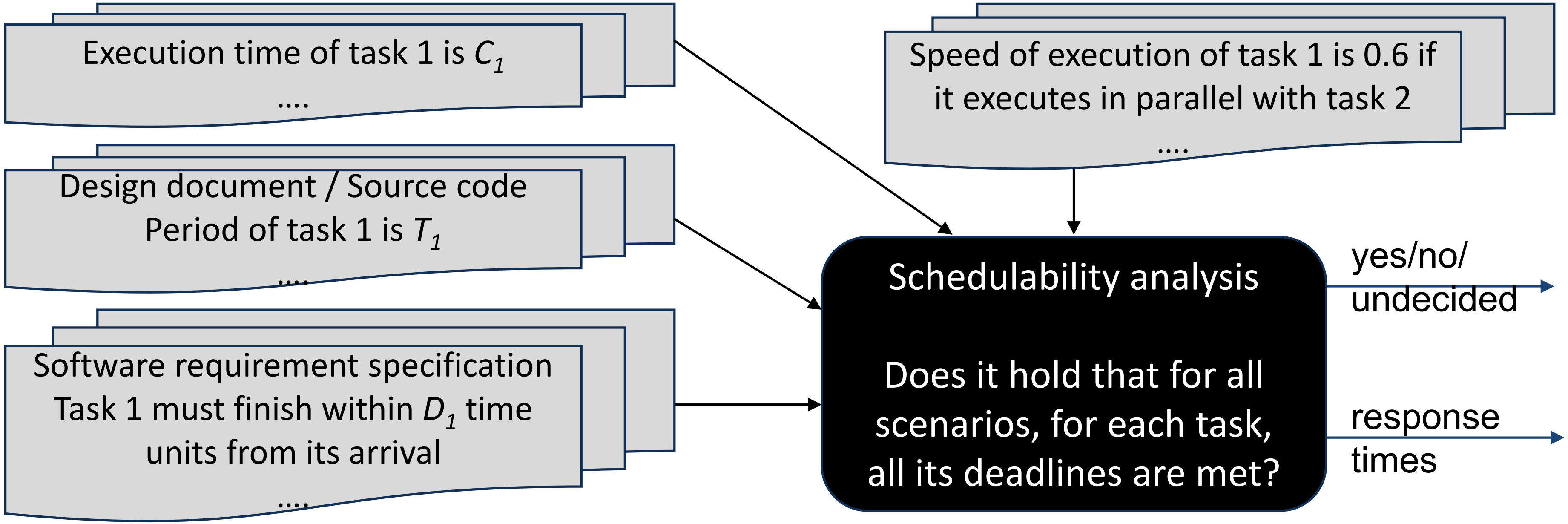


Overview of Our Solution



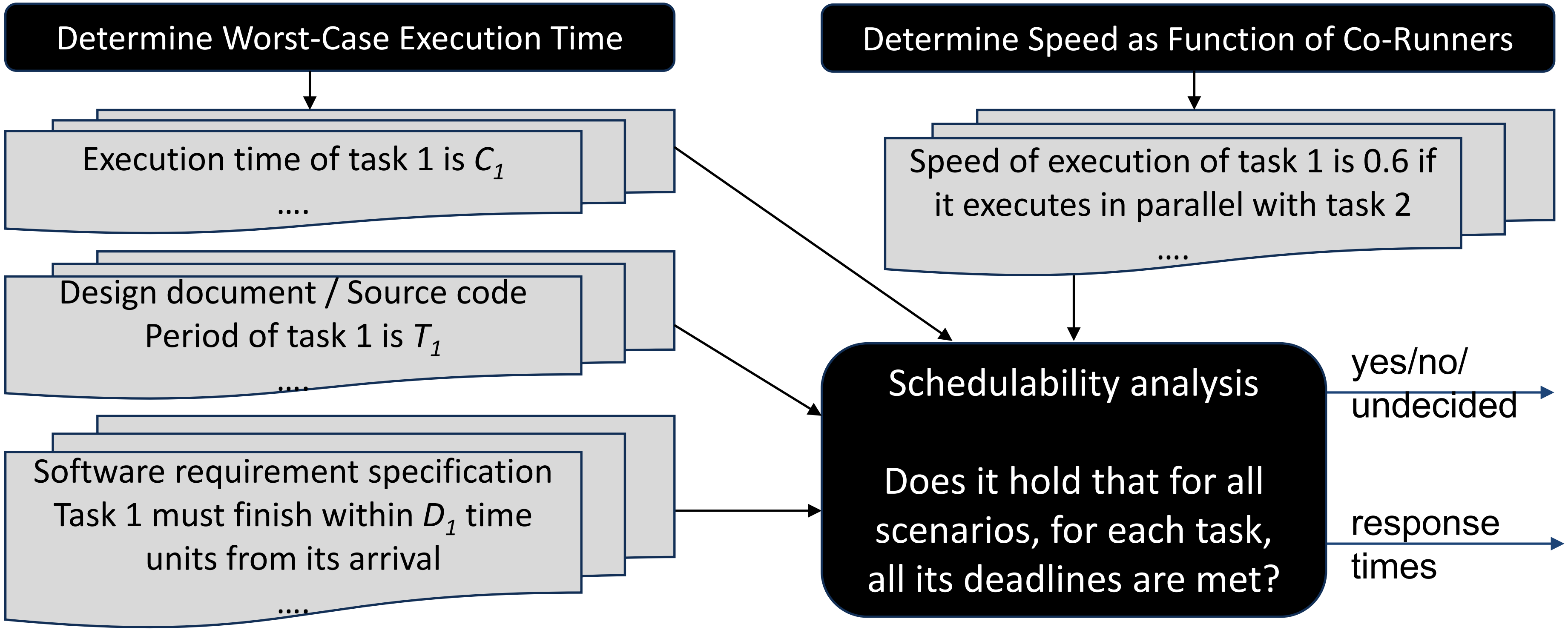
Carnegie Mellon University Software Engineering Institute

Overview of Our Solution

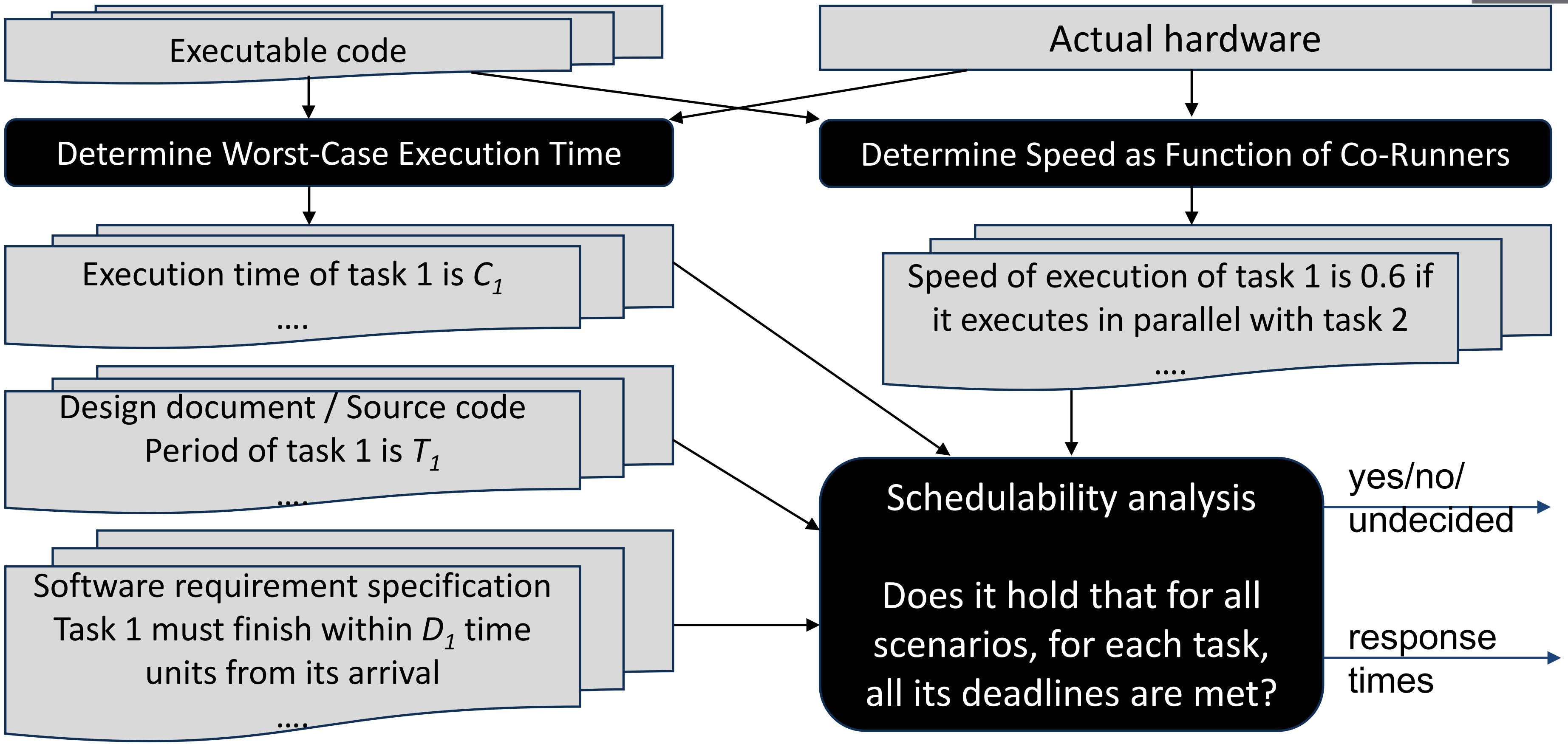


Carnegie Mellon University Software Engineering Institute

Overview of Our Solution



Overview of Our Solution



Carnegie Mellon University Software Engineering Institute

Our Tools Perform These Tasks

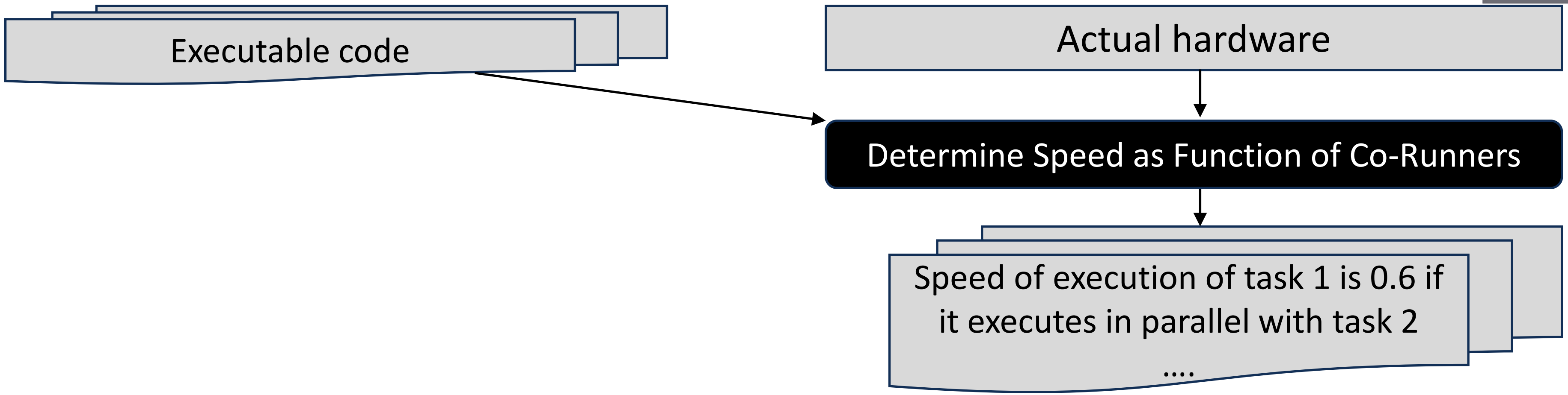
Determine Worst-Case Execution Time

Determine Speed as Function of Co-Runners

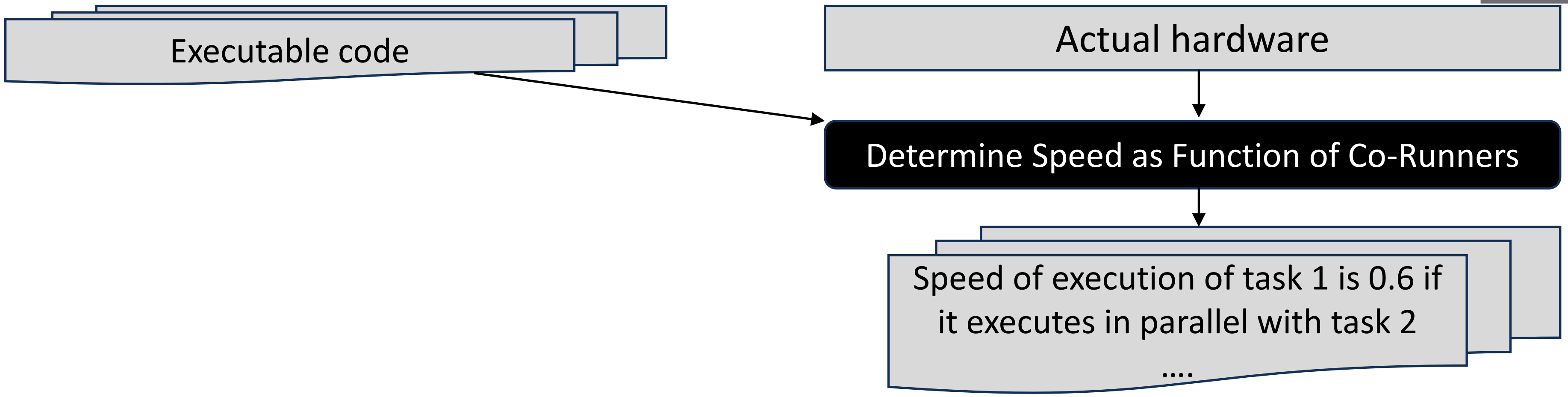
Schedulability analysis

Does it hold that for all scenarios, for each task, all its deadlines are met?

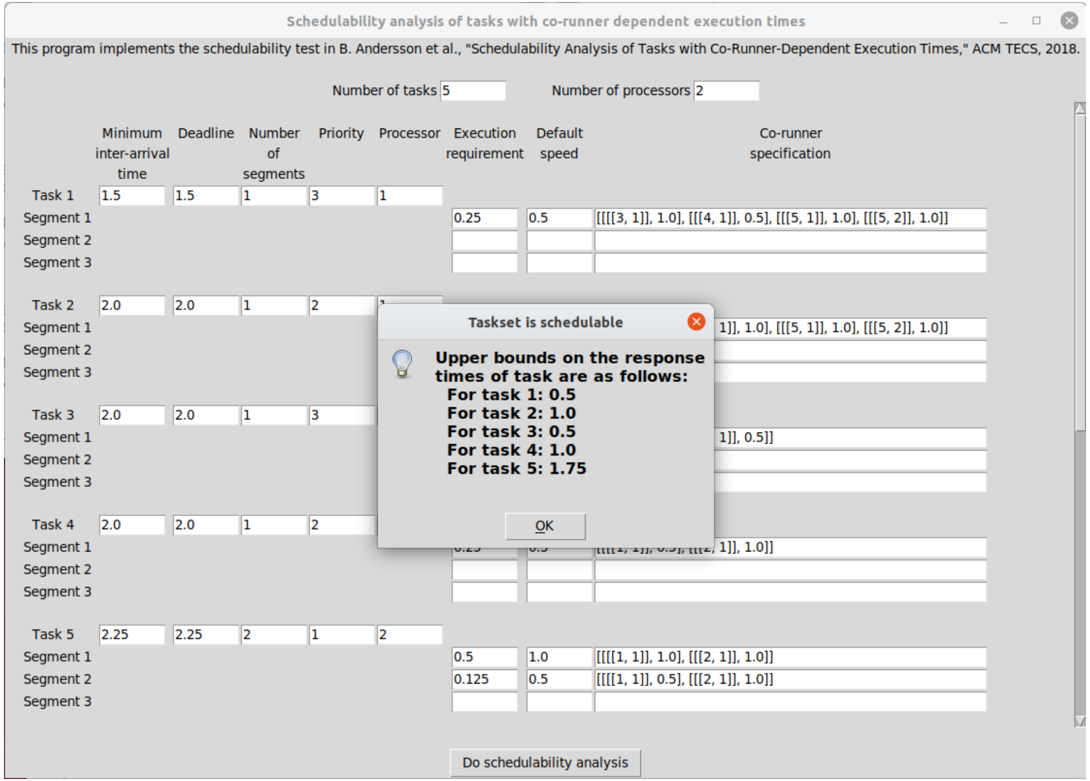
Our Abstraction Speed as Function of Co-Runners Is New



Our Abstraction Speed as Function of Co-Runners Allows undoc hw



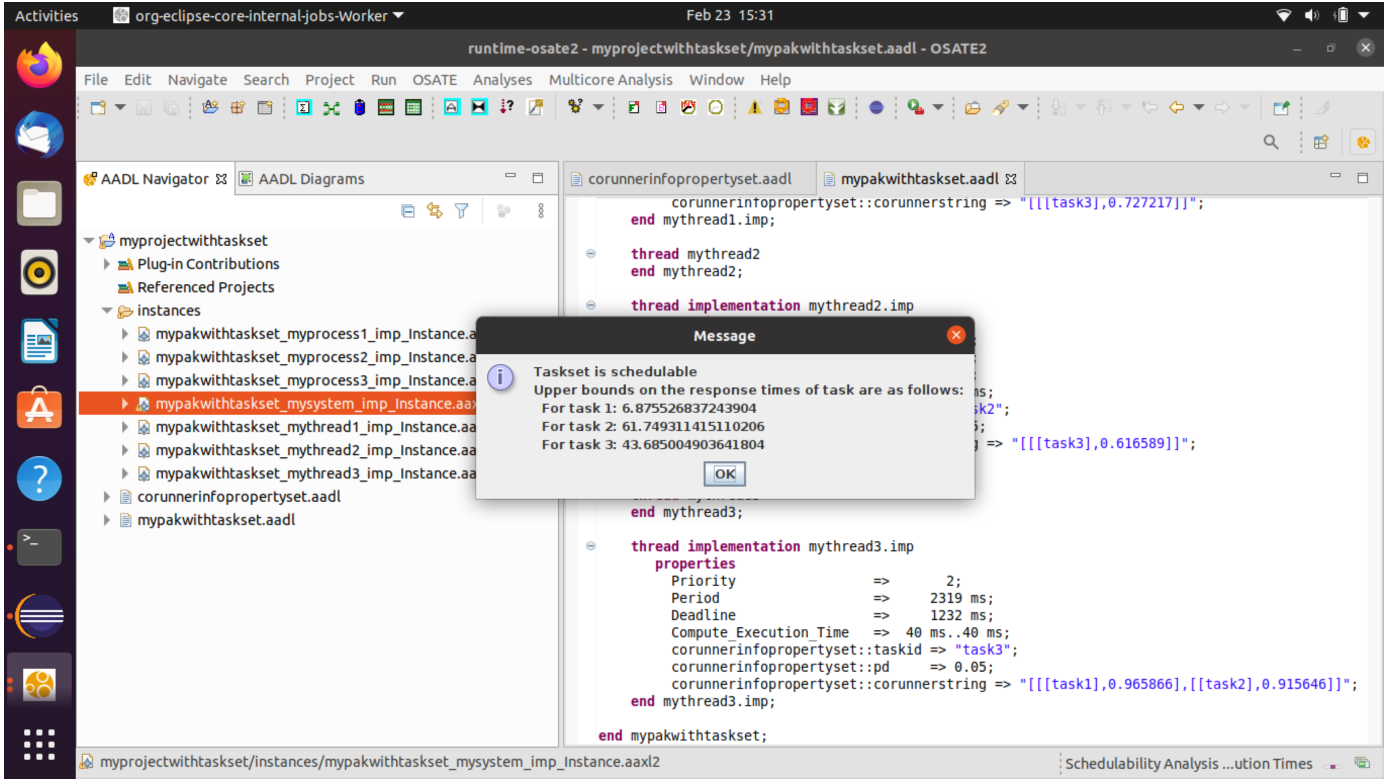
A Look at Our Tools - 1



Main Idea

Fixed-point iteration where each iteration solves a linear program

A Look at Our Tools - 2



A Look at Our Tools - 3

Carnegie Mellon University Software Engineering Institute

```
ba@ba-desktop: ~/ga_find_wcet
File Edit View Search Terminal Help
ba@ba-desktop:~/ga_find_wcet$ more myconf
0
262144

ba@ba-desktop:~/ga_find_wcet$ ./ga_find_wcet myconf ./bubblesort
0.306359
ba@ba-desktop:~/ga_find_wcet$ time ./bubblesort < ascending_integers.dat

real    0m0.002s
user    0m0.001s
sys     0m0.001s
ba@ba-desktop:~/ga_find_wcet$ time ./bubblesort < descending_integers.dat

real    0m0.275s
user    0m0.275s
sys     0m0.000s
ba@ba-desktop:~/ga_find_wcet$ time ./bubblesort < worstcaseinput_GA_inputfile.dat

real    0m0.308s
user    0m0.308s
sys     0m0.000s
ba@ba-desktop:~/ga_find_wcet$
```

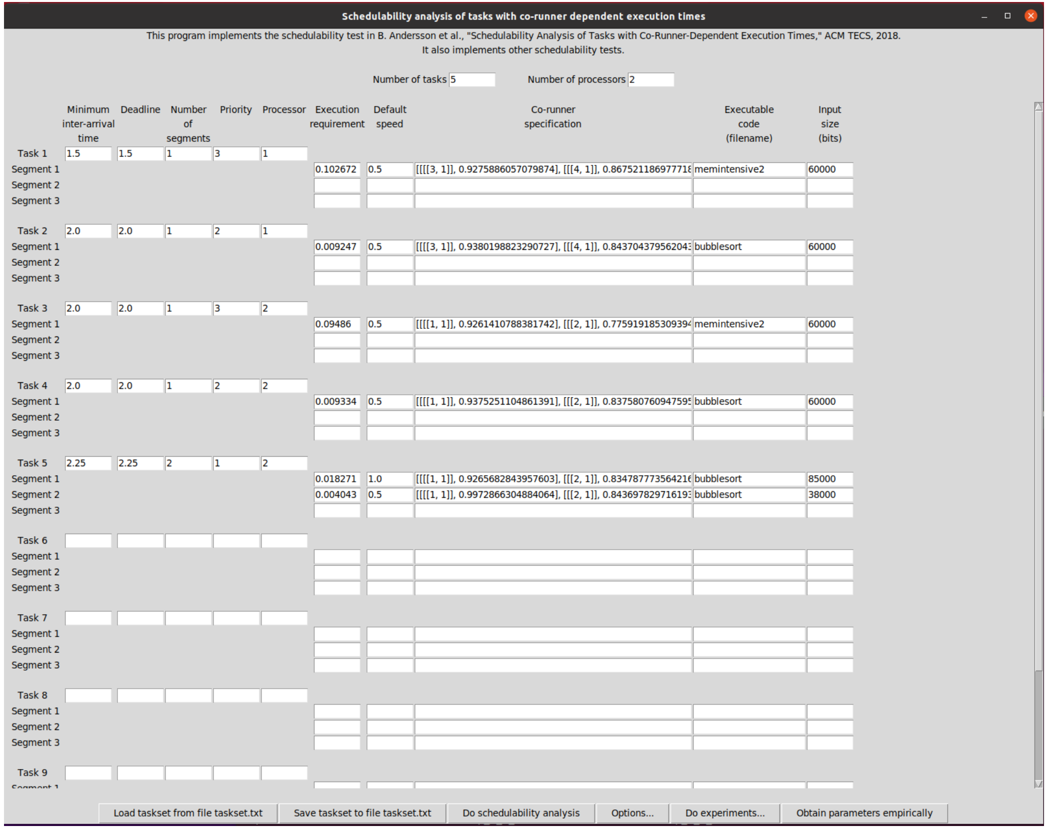
Main Idea

Use genetic algorithms

Fitness = execution time
Gene = input to program

Maximize fitness
=
Maximize execution time

A Look at Our Tools - 4



Main Idea

Use genetic algorithms

Fitness = slowdown
Gene = input to program

Maximize fitness
=
Maximize slowdown

Publications

Andersson, B.; Kim, H.; de Niz, D.; Klein, M.; Rajkumar, R.; & Lehoczky, J. Schedulability Analysis of Tasks with Co-Runner-Dependent Execution Times. *ACM Transactions on Embedded Computing Systems*. Volume 17. Number 3. Article 71. May 2018.

Andersson, B.; de Niz, D.; & Klein, M. *The Multicore Challenge in Assured Autonomy*. Presented at AUVSI Exponential. August 2021.

Kim, H.; de Niz, D.; Andersson, B.; Klein, M.; & Lehoczky, J. *Addressing Multi-Core Timing Interference using Co-Runner Locking*. To be presented at IEEE Real-Time Systems Symposium. December 2021.

Petrucci, V.; Andersson, B.; Massa, E.; & Lima, G. *Heterogeneous Quasi-Partitioned Scheduling*. To be presented at IEEE Real-Time Systems Symposium. December 2021.

Tools

http://www.andrew.cmu.edu/user/banderss/projects.html

Transition success

Tools have been used in internal seminars at Army CCDC Aviation & Missile Center (AvMC)

Team Members

Bjorn Andersson
Principal Researcher
Software Solutions Division
Software Engineering Institute

Dionisio Di Niz
Technical Director, Assuring CyberPhysical Systems
Software Solutions Division
Software Engineering Institute

Mark Klein
Principal Technical Advisor
Software Solutions Division
Software Engineering Institute

John Lehoczky
Thomas Lord University Professor of Statistics
Carnegie Mellon University

Hyoseung Kim
Associate Professor
Department of Electrical and Computer Engineering
University of California, Riverside

Bill Anderson
Senior Member of the Technical Staff (retired)
Software Solutions Division
Software Engineering Institute

Anton Dimov Hristozov
Software Engineer
Software Solutions Division
Software Engineering Institute

Contact Information

Carnegie Mellon University Software Engineering Institute

Point of Contact

Bjorn Andersson, Ph.D.

Principal Researcher

Telephone: +1 412.268.9243

Email: info@sei.cmu.edu

Carnegie Mellon University Software Engineering Institute

Thanks!