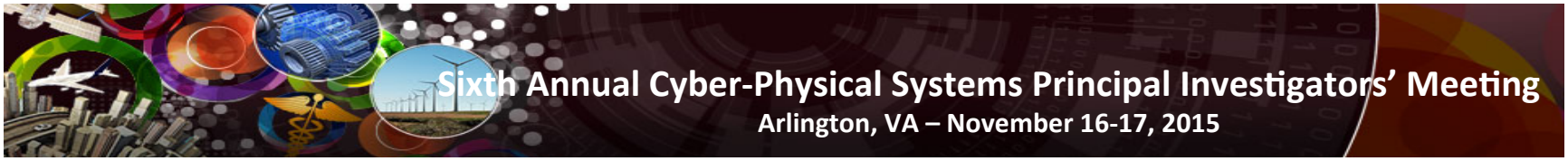


Enabling the Usage of Multicore Platforms in Real-Time Safety-Critical Systems

James H. Anderson
Kenan Professor, UNC Chapel Hill

CPS 1239135

CPS: Breakthrough: Collaborative Research: Bringing the Multicore Revolution to Safety-Critical Cyber-Physical Systems

PI: James H. Anderson, UNC Chapel Hill

co-PI: Frank Mueller, NC State University

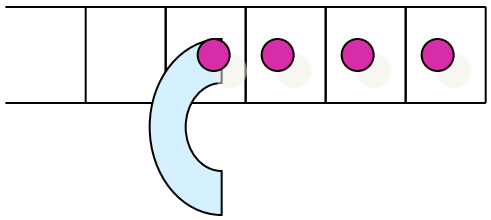
Outline

- Real-time 101.
- Problems caused by multicore.
  - "The one-out-of-m problem."
  - Why this is an important problem.
- Solution strategy.
- Evaluation.
- Project summary.

CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

What is a Real-Time System?

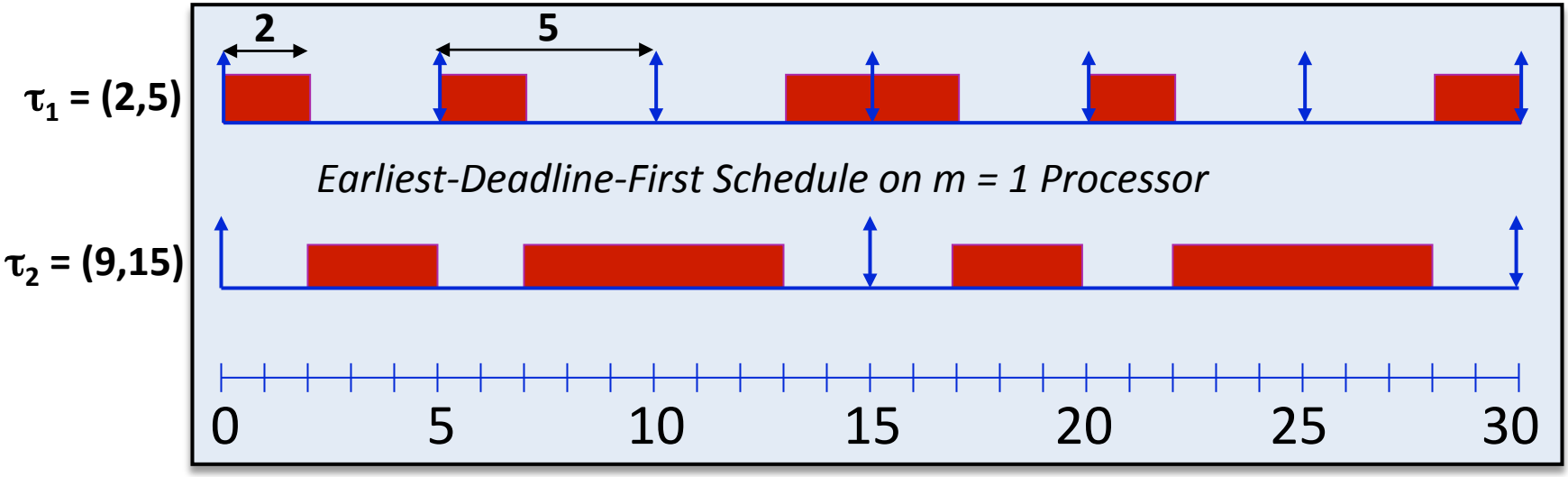
- A system with a dual notion of correctness:
  - *Logical* correctness ("it does the right thing");
  - *Temporal* correctness ("it does it on time").
- A system wherein *predictability* is as important as *performance*.
- **A Simple Example:** A robot arm picking up objects from a conveyor belt.



CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

Representing Real-Time Workloads: Periodic Task Systems

- Consider a set τ of n periodic tasks on m processors:

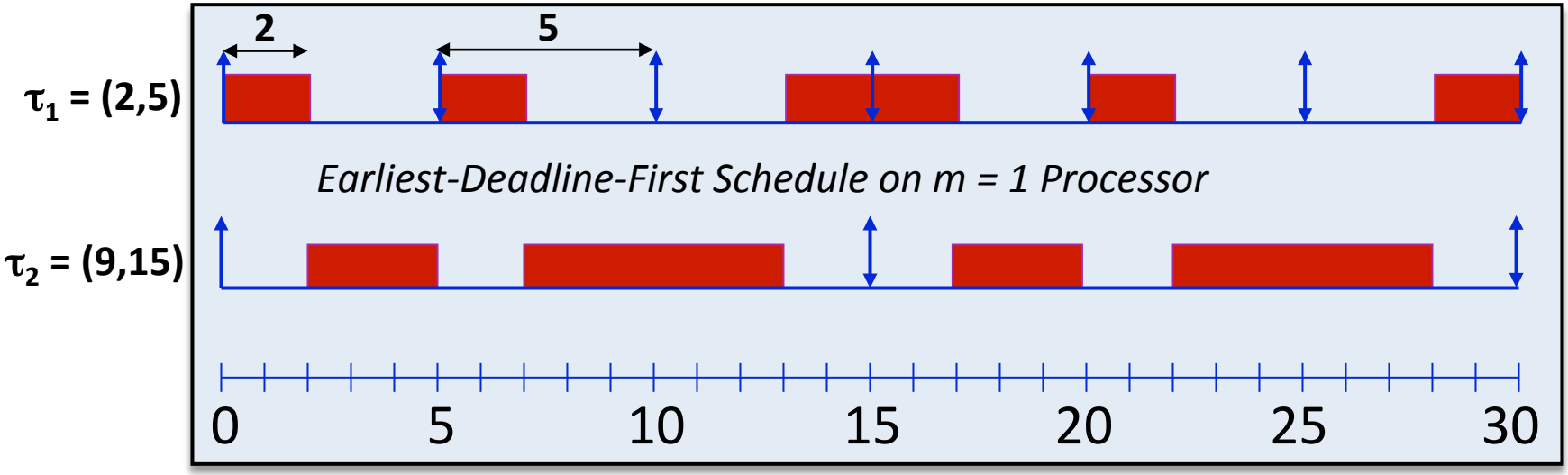


CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

job release ↓ job deadline

Representing Real-Time Workloads: Periodic Task Systems

- Consider a set τ of n periodic tasks on m processors:
  - Each task $\tau_i = (C_i, T_i)$ releases a *job* with worst-case execution time (WCET) C_i at least T_i time units apart.
    - τ_i's *utilization* is $U_i = C_i/T_i$.
    - *Total utilization* is $U(\tau) = \sum_i C_i/T_i$.

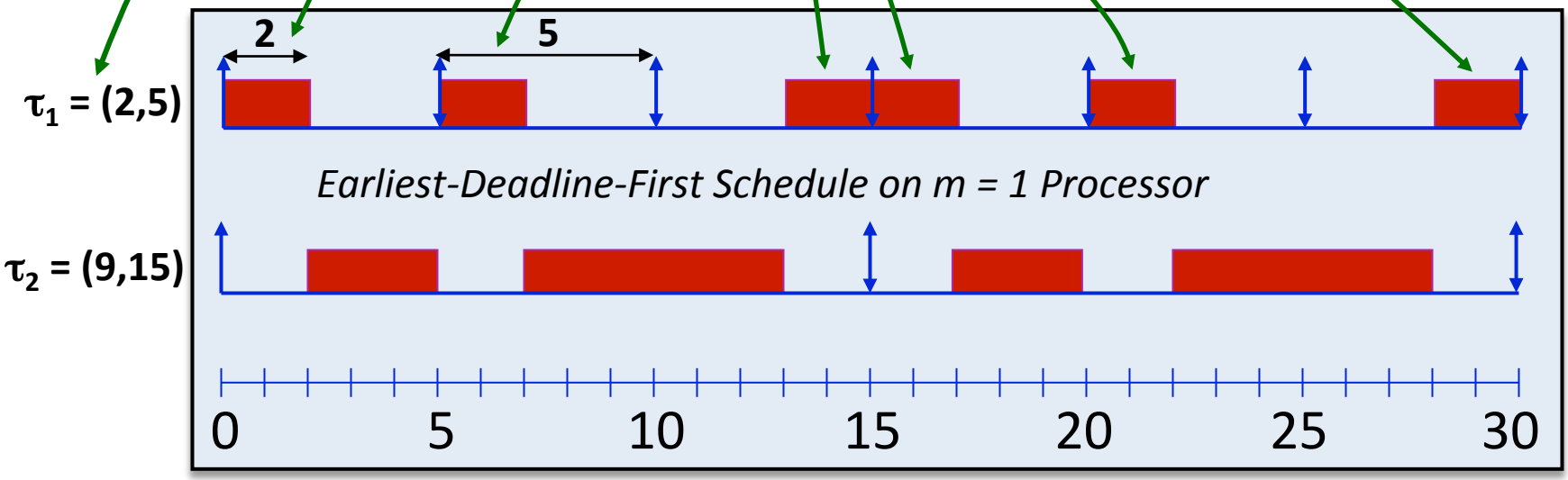


CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

job release ↓ job deadline

Representing Real-Time Workloads: Periodic Task Systems

- Consider a set τ of n periodic tasks on m processors:
  - Each task $\tau_i = (C_i, T_i)$ releases a *job* with worst-case execution time (WCET) C_i at least T_i time units apart.
    - τ_i's *utilization* is $U_i = C_i/T_i$.
    - *Total utilization* is $U(\tau) = \sum_i C_i/T_i$.

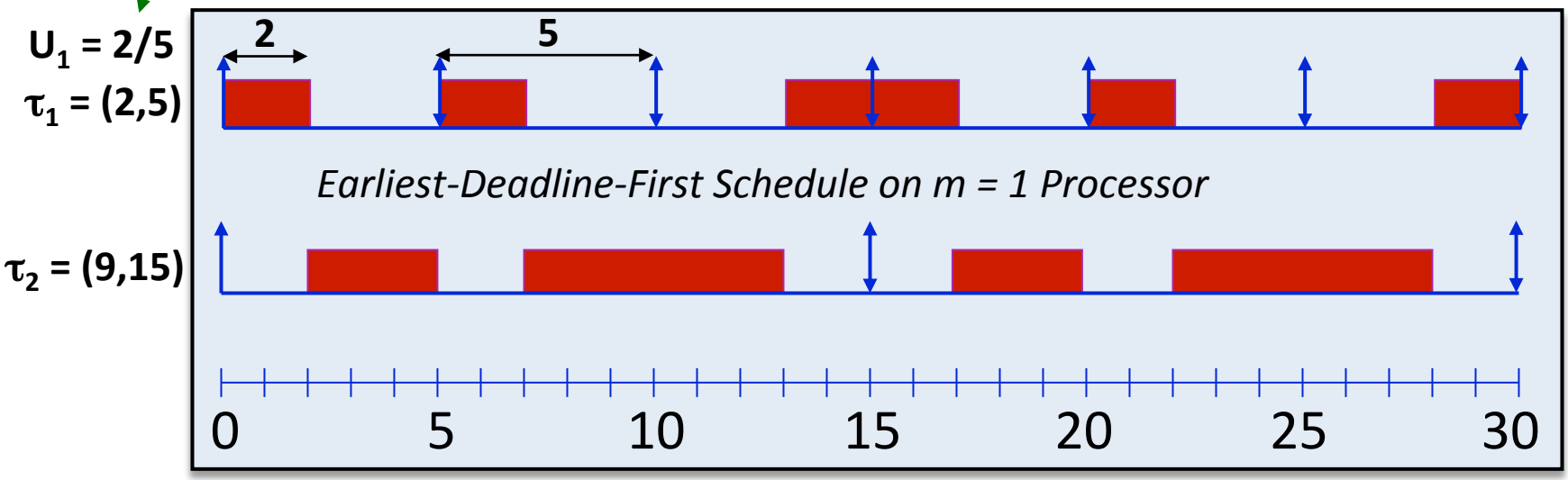


CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

job release ↓ job deadline

Representing Real-Time Workloads: Periodic Task Systems

- Consider a set τ of n periodic tasks on m processors:
  - Each task $\tau_i = (C_i, T_i)$ releases a *job* with worst-case execution time (WCET) C_i at least T_i time units apart.
    - τ_i's *utilization* is $U_i = C_i/T_i$.
    - *Total utilization* is $U(\tau) = \sum_i C_i/T_i$.

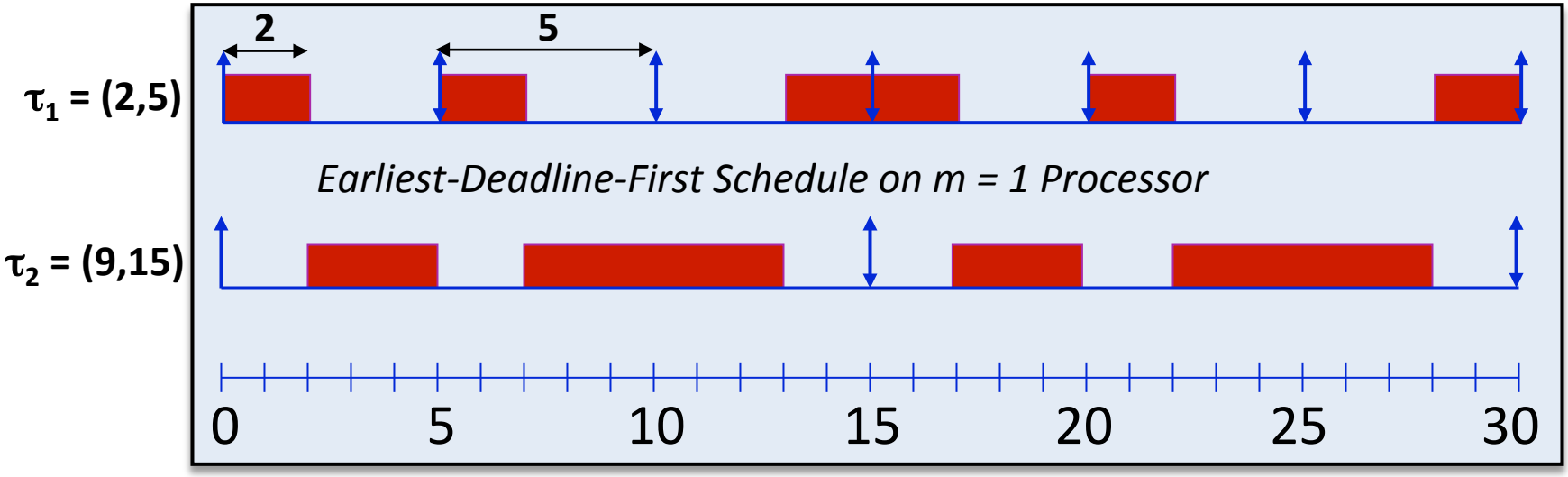


CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

job release ↓ job deadline

Representing Real-Time Workloads: Periodic Task Systems

- Consider a set τ of n periodic tasks on m processors:
  - Each task $\tau_i = (C_i, T_i)$ releases a *job* with worst-case execution time (WCET) C_i at least T_i time units apart.
    - τ_i's *utilization* is $U_i = C_i/T_i$.
    - *Total utilization* is $U(\tau) = \sum_i C_i/T_i$.
  - Each job of τ_i has a *relative deadline* given by T_i.

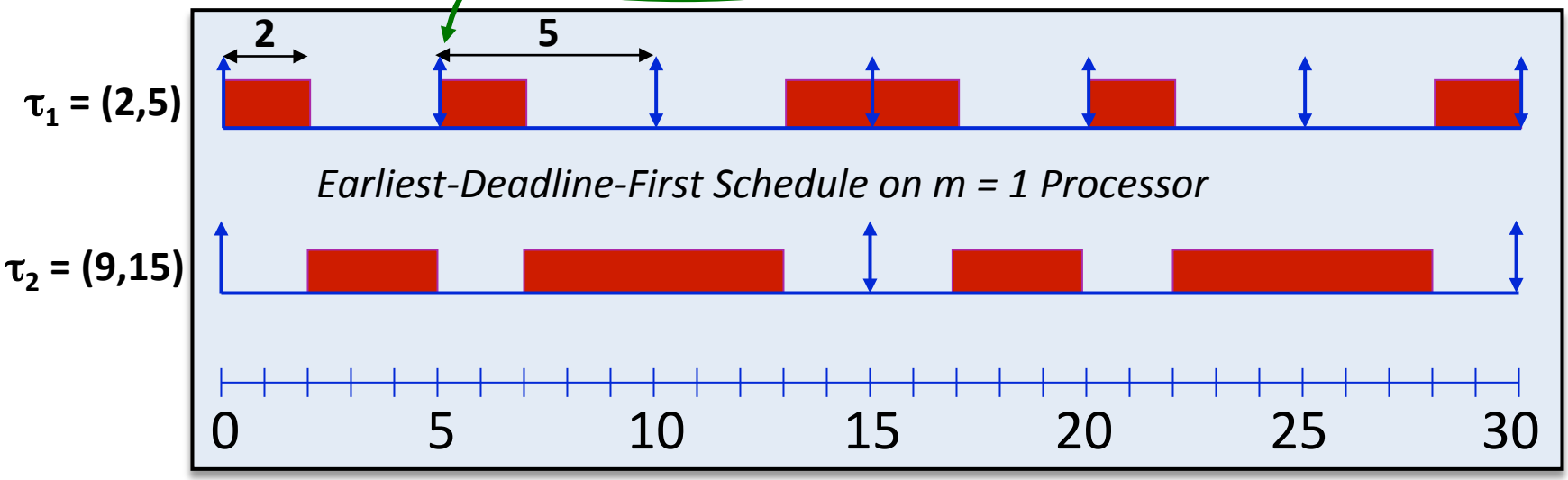


CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

job release ↓ job deadline

Representing Real-Time Workloads: Periodic Task Systems

- Consider a set τ of n periodic tasks on m processors:
  - Each task $\tau_i = (C_i, T_i)$ releases a *job* with worst-case execution time (WCET) C_i at least T_i time units apart.
    - τ_i's *utilization* is $U_i = C_i/T_i$.
    - *Total utilization* is $U(\tau) = \sum_i C_i/T_i$.
  - Each job of τ_i has a *relative deadline* given by T_i.



CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

job release ↓ job deadline

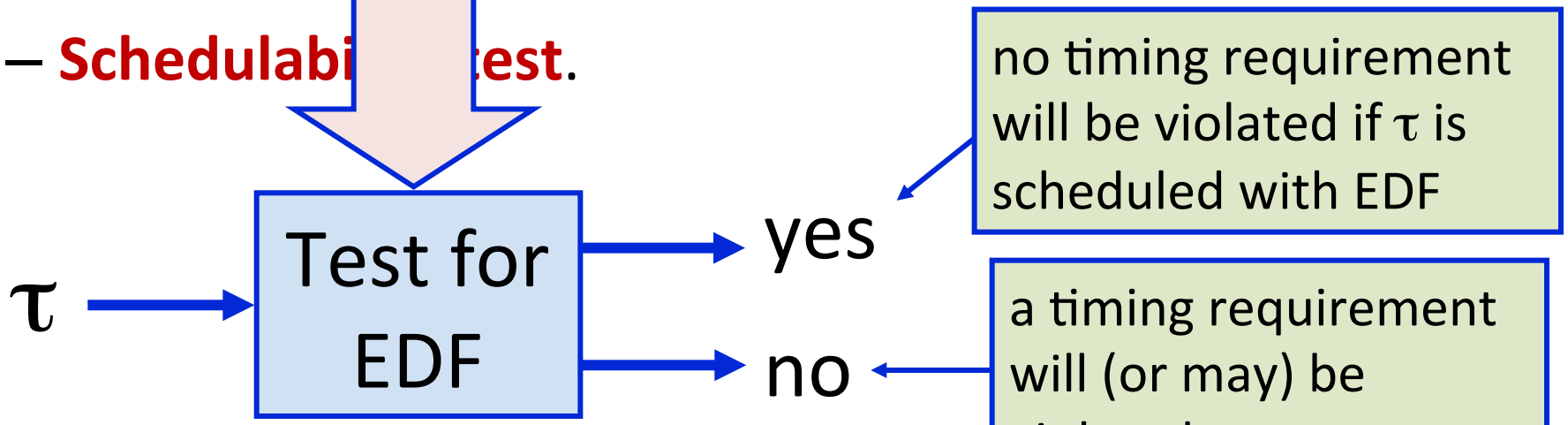
Checking Timing Constraints: Schedulability Tests

- W.r.t. scheduling, we actually care about *two* kinds

Capacity loss occurs when test requires conservative assumptions about task execution times, the manner in which tasks delay each other, etc.

deadlines higher priority.

– **Schedulabi est**.



CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

Outline

- Real-time 101.
- Problems caused by multicore.
  - "The one-out-of-m problem."
  - Why this is an important problem.
- Solution strategy.
- Evaluation.
- Project summary.

CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

Schedulability Problems Caused by Multicore

- With multicore, the state of the art today in many safety-critical domains such as avionics is to ***disable all but one core***.
- Why?
  - On a machine with m cores, analysis pessimism can easily cause capacity loss of m – 1 or greater.
  - We call this the **"one-out-of-m" problem**:

    Given m cores, the capacity of the additional m – 1 cores is wasted.

  - *This is the most important open problem concerning "real-time on multicore" today.*

CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

The Root of the Problem

- This problem is caused by the presence of shared hardware (caches, buses, memory banks, etc.) that is not predictably managed.
- See the FAA position paper CAST 32 for an extensive discussion of problems caused by shared-hardware interference.

CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

Outline

- Real-time 101.
- Problems caused by multicore.
  - "The one-out-of-m problem."
  - Why this is an important problem.
- Solution strategy.
- Evaluation.
- Project summary.

CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

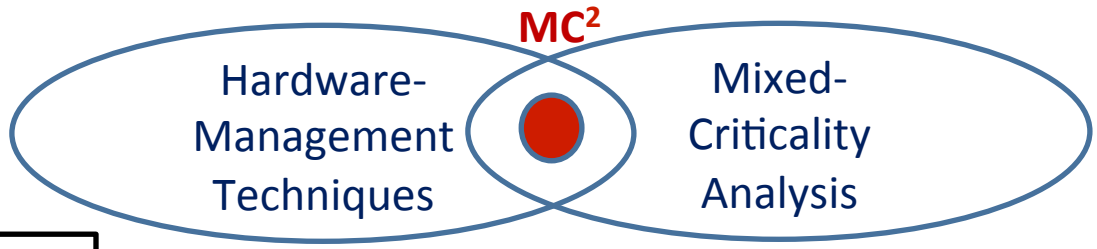
Our Solution Strategy

- W.r.t. lessening capacity loss generally (even on uniprocessors), two orthogonal approaches have been investigated previously:
  - **Hardware-management techniques** that reduce hardware interference.
  - **Mixed-criticality analysis techniques** that enable less critical tasks to be provisioned less pessimistically.

CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

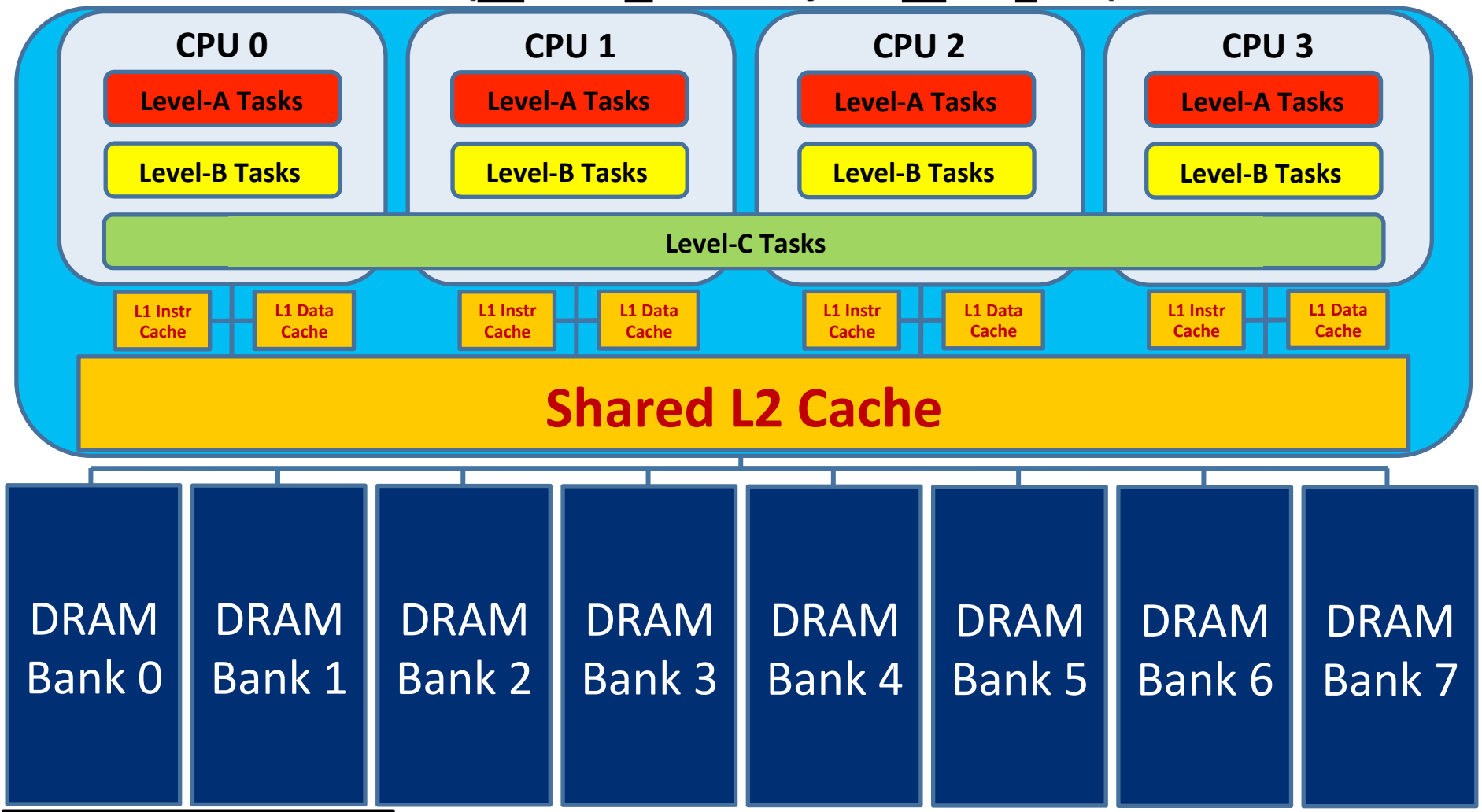
Our Solution Strategy

- Our work focuses broadly on research questions that arise when applying both approaches together.
  - *Can better platform utilization be realized if hardware resources are managed differently at different criticality levels?*
  - *If so, how should resources be managed both within and across criticality levels?*
  - We are addressing such questions in the context of a resource-allocation and analysis framework developed by us called **MC² (mixed criticality on multicore).**



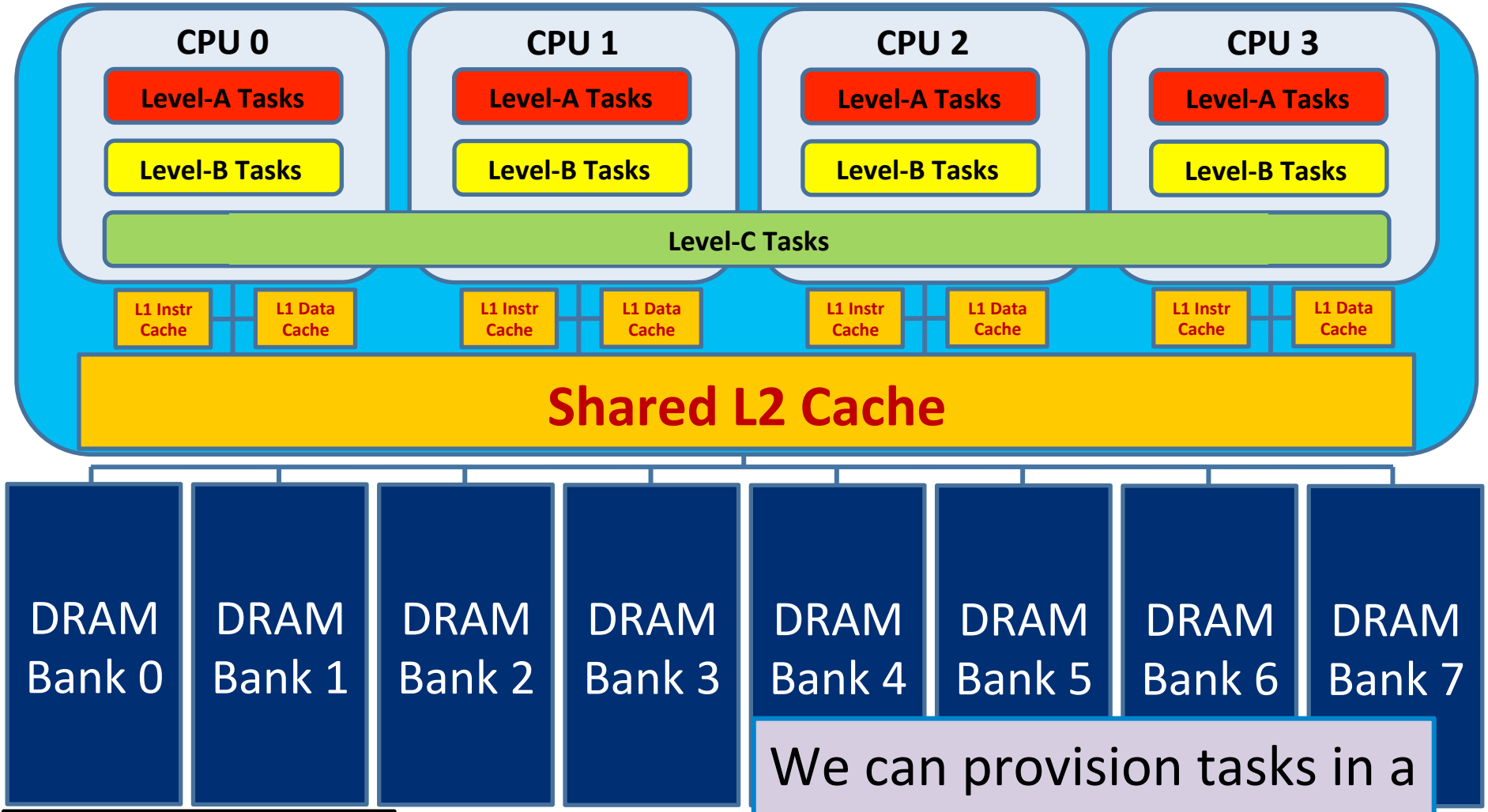
CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

MC² (Mixed-Criticality on Multicore)



CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

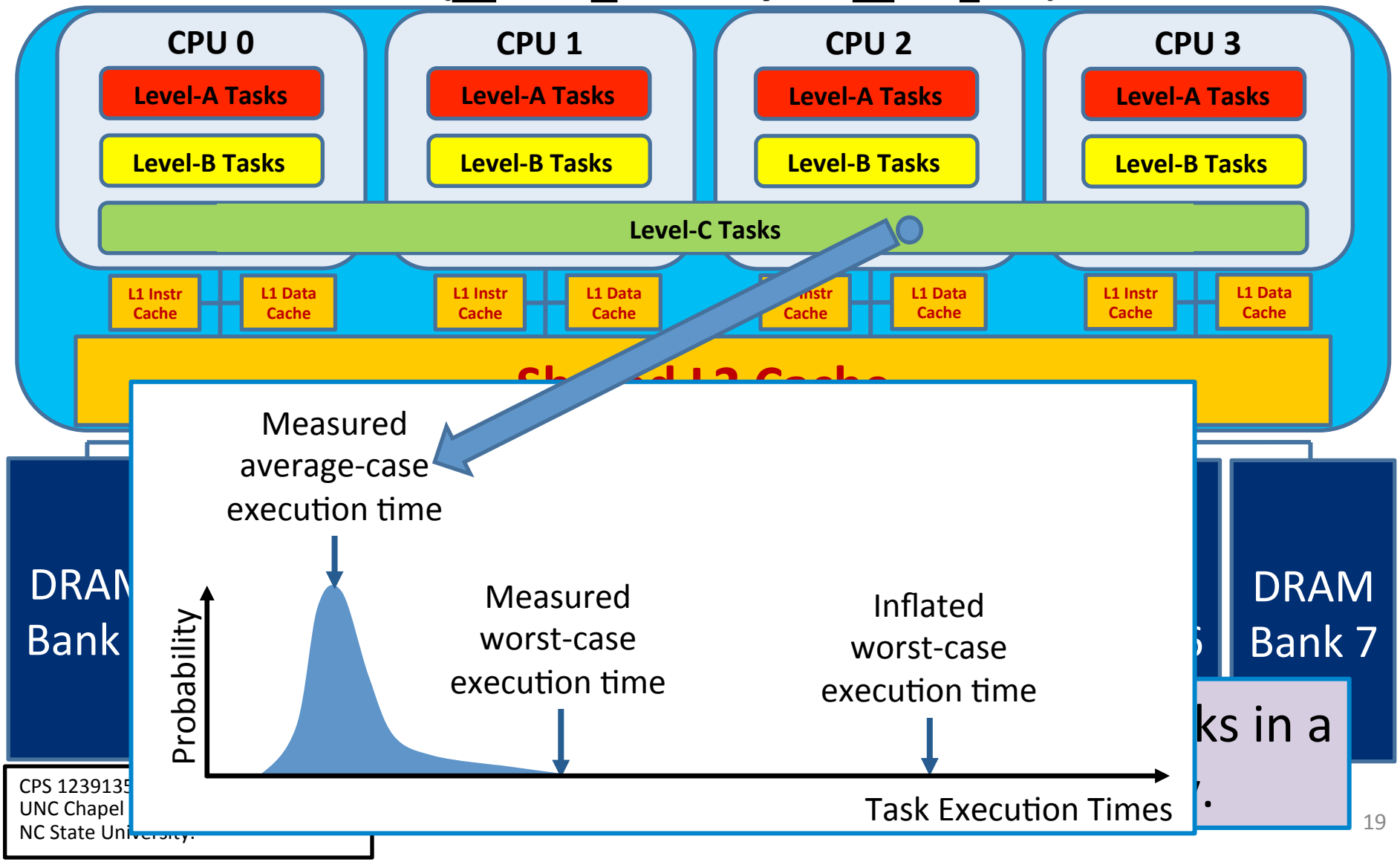
MC² (Mixed-Criticality on Multicore)



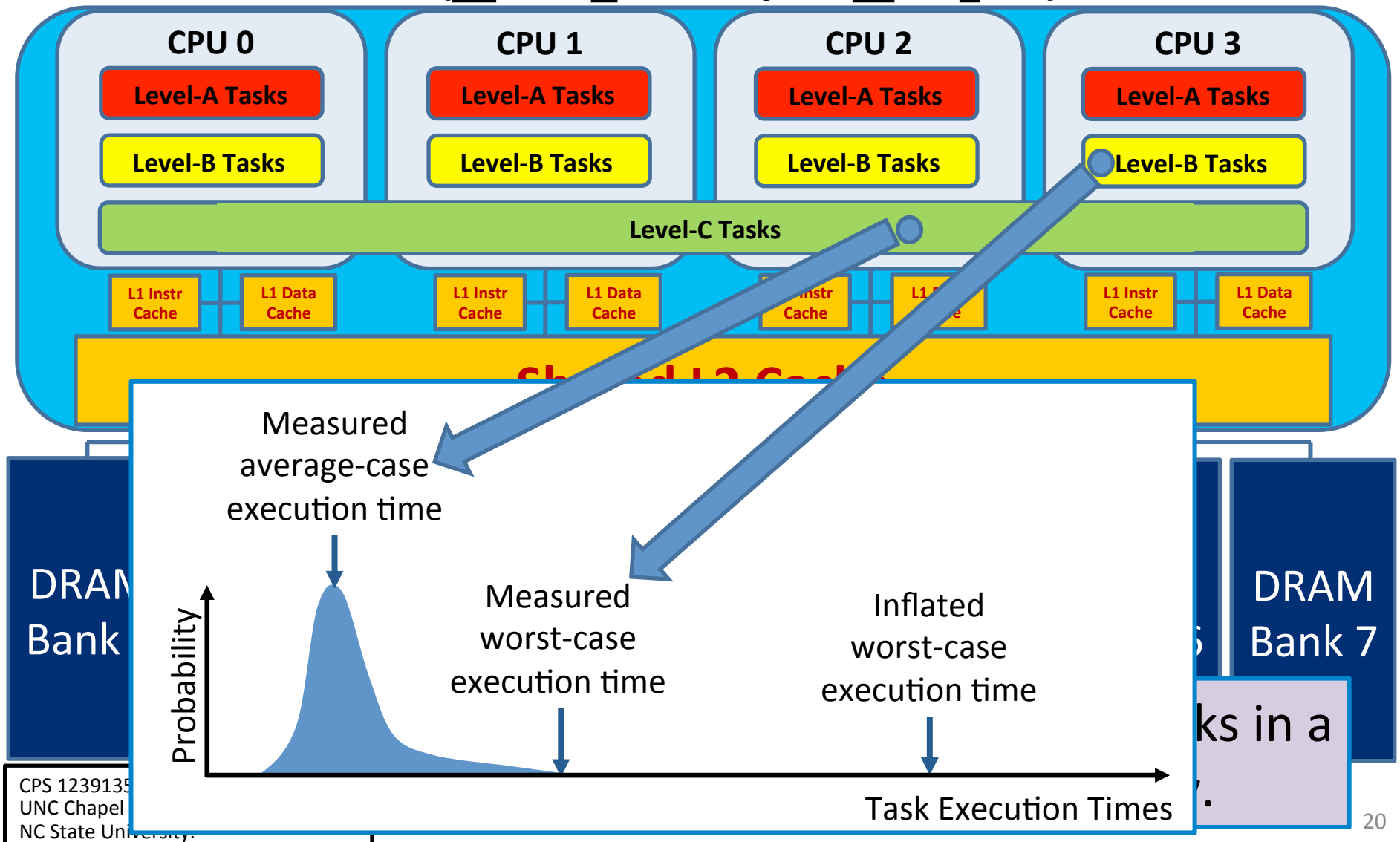
We can provision tasks in a **criticality-aware** way.

CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

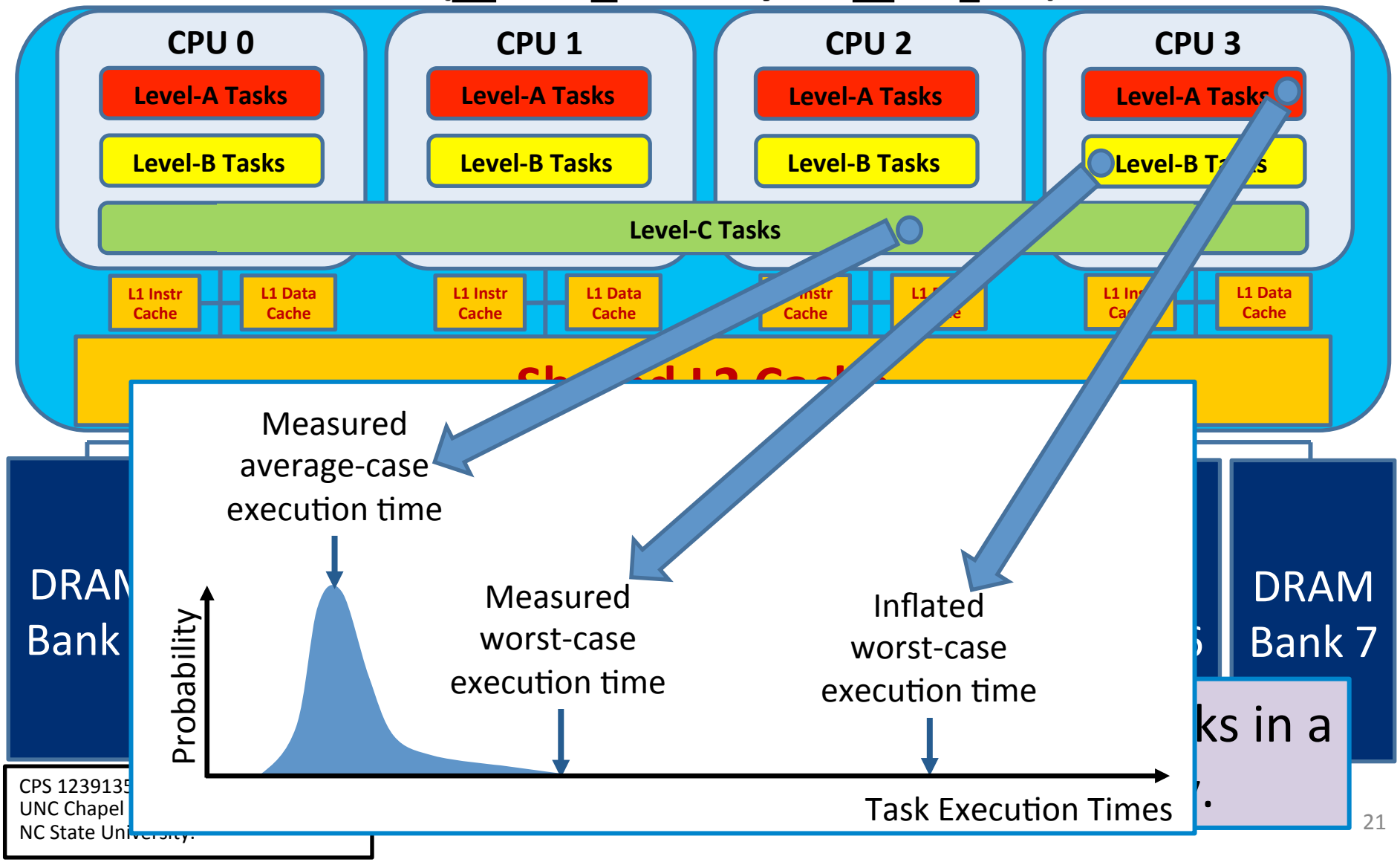
MC² (Mixed-Criticality on Multicore)



MC² (Mixed-Criticality on Multicore)

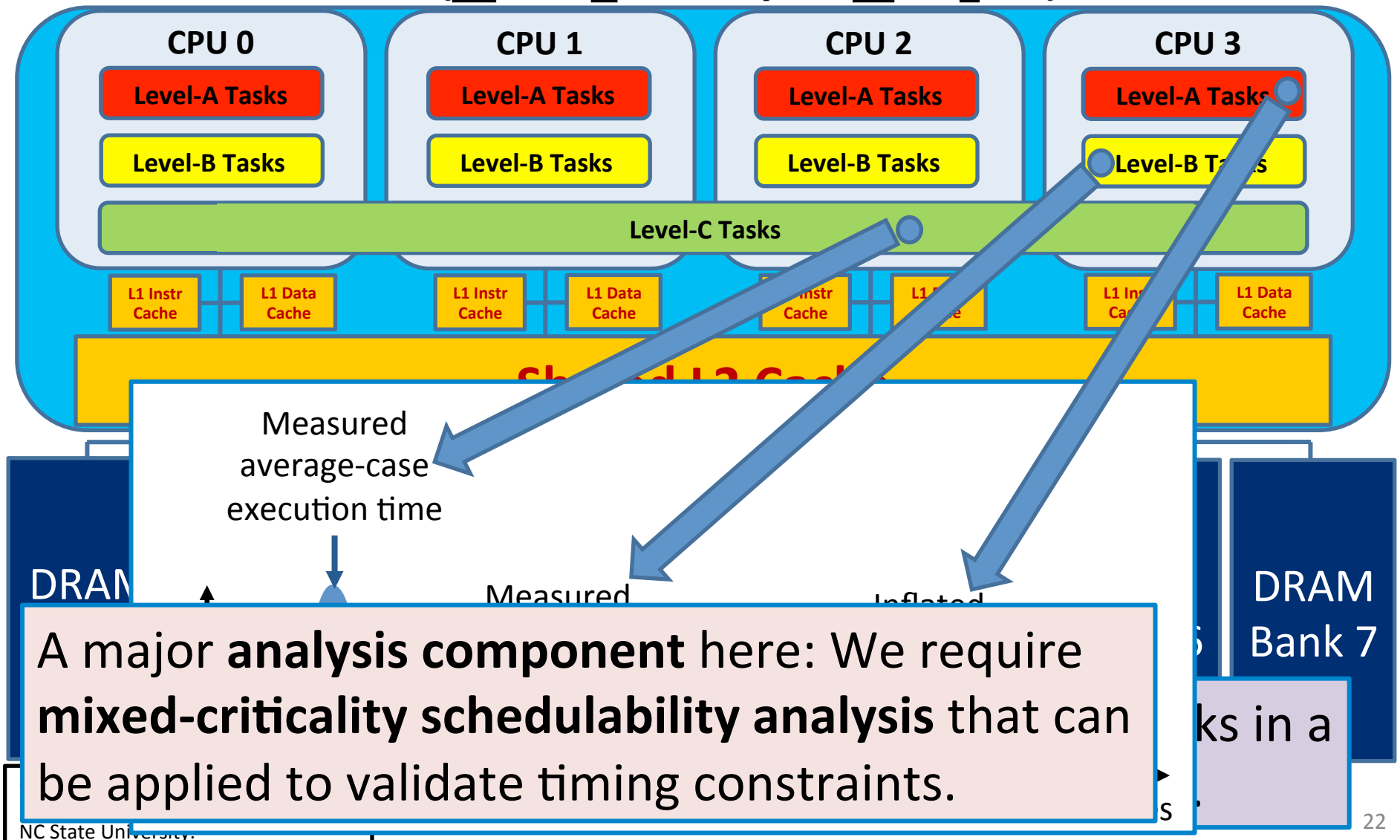


MC² (Mixed-Criticality on Multicore)



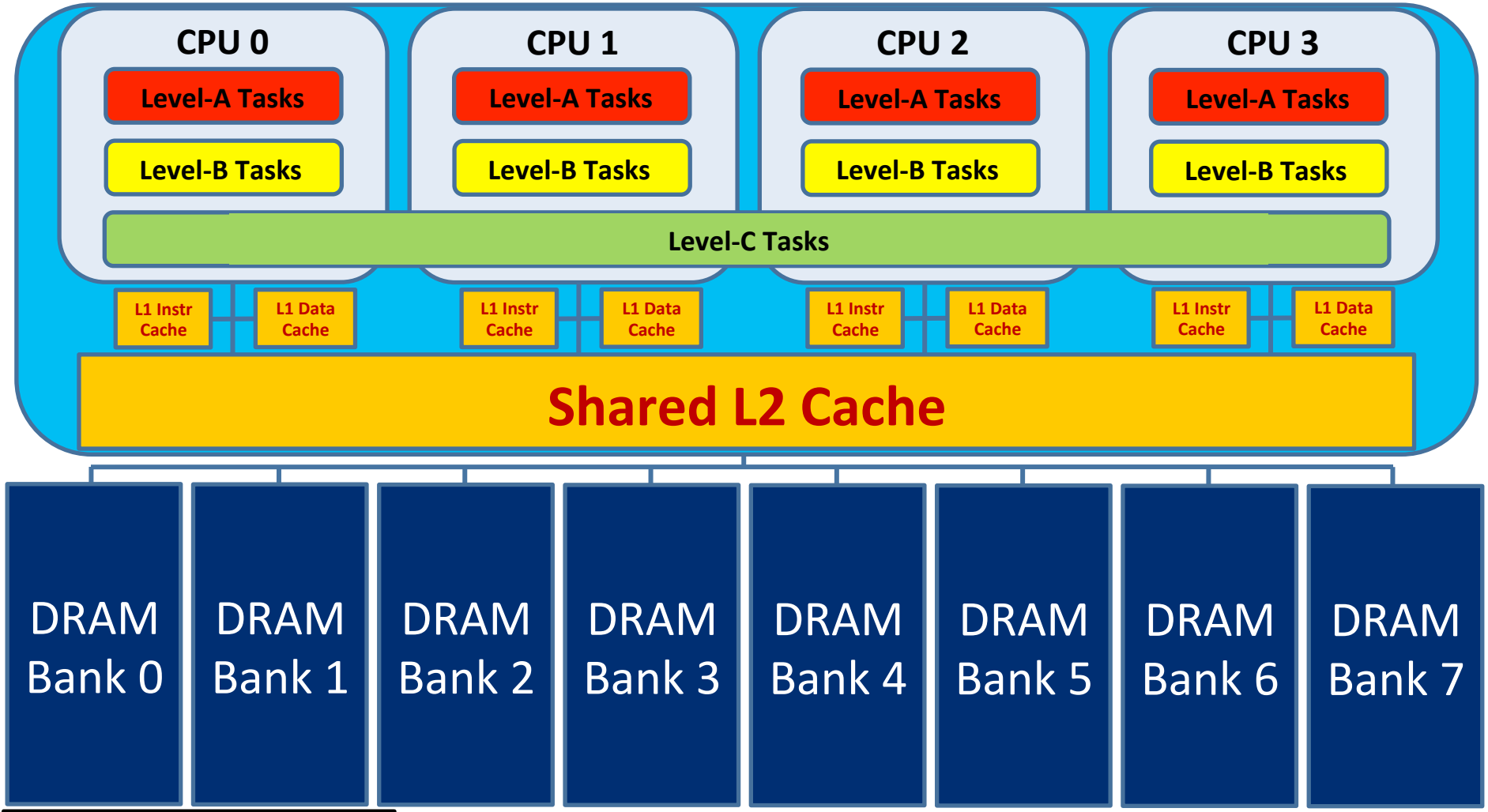
CPS 1239135
UNC Chapel
NC State University

MC² (Mixed-Criticality on Multicore)



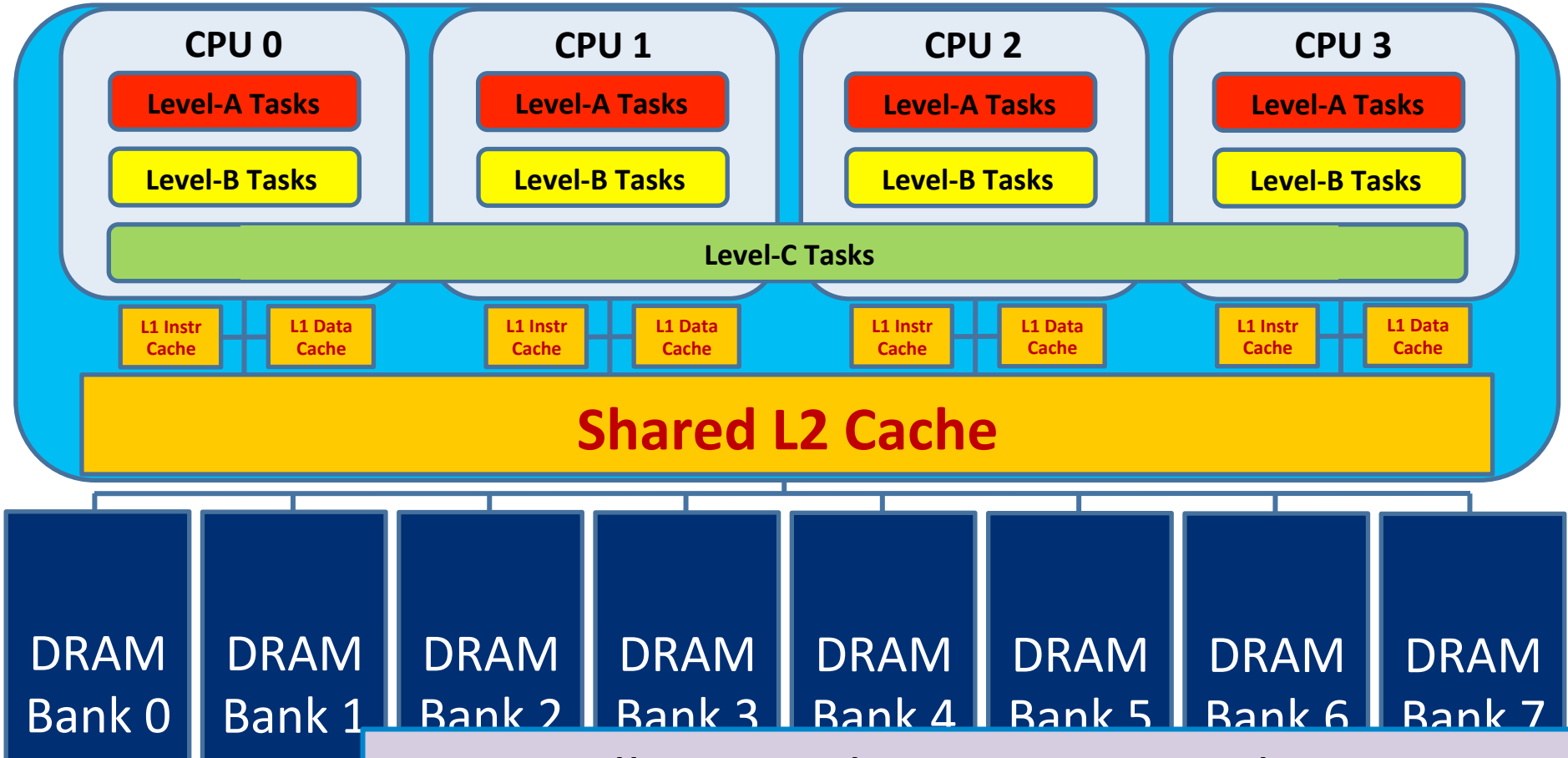
A major **analysis component** here: We require **mixed-criticality schedulability analysis** that can be applied to validate timing constraints.

MC² (Mixed-Criticality on Multicore)



CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

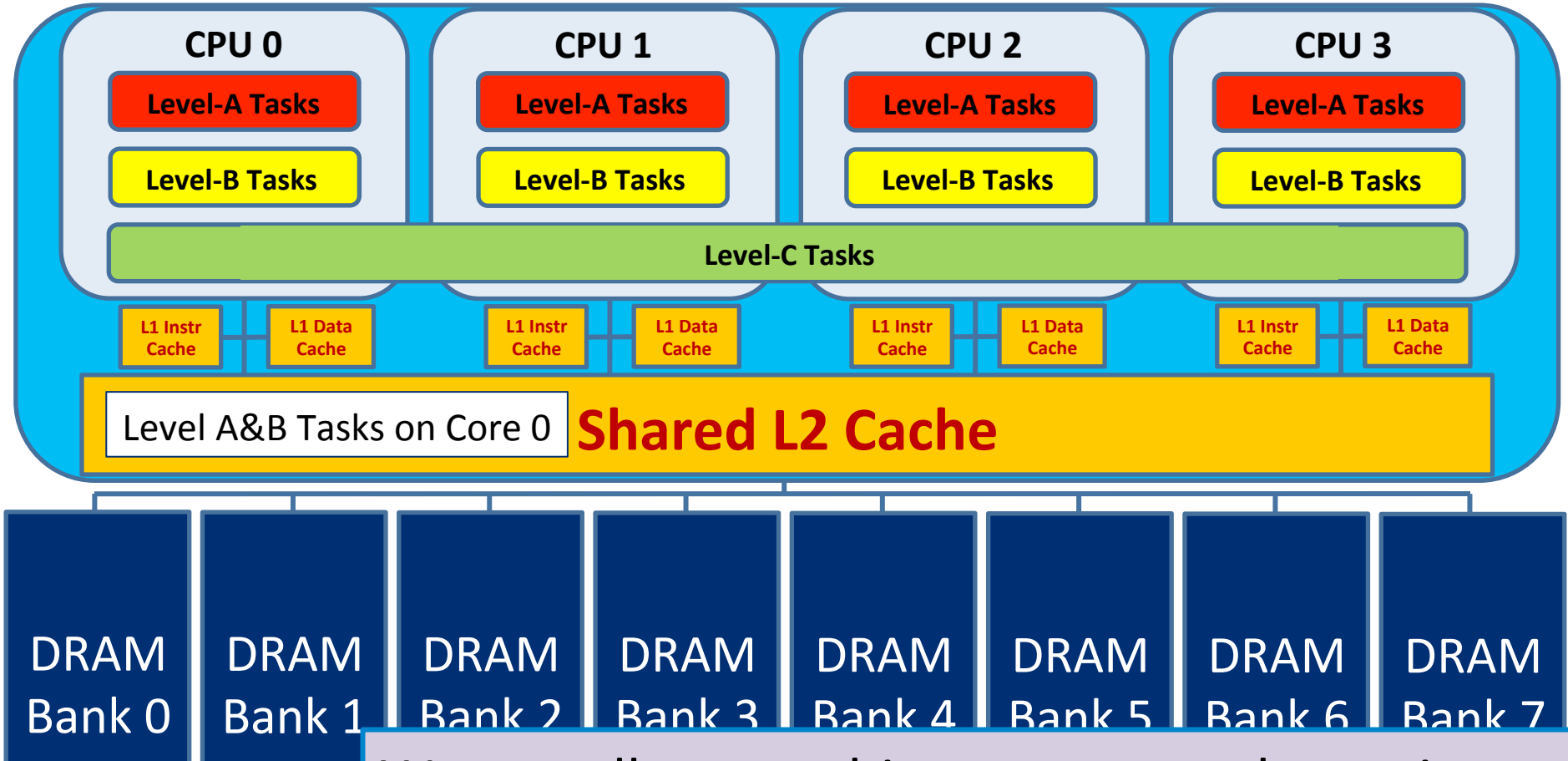
MC² (Mixed-Criticality on Multicore)



We can allocate arbitrary rectangular regions of the **shared L2 cache** to sets of tasks.

CPS 1239135, PI: James H. Anderso
UNC Chapel Hill, co-PI: Frank Mueller,
NC State University.

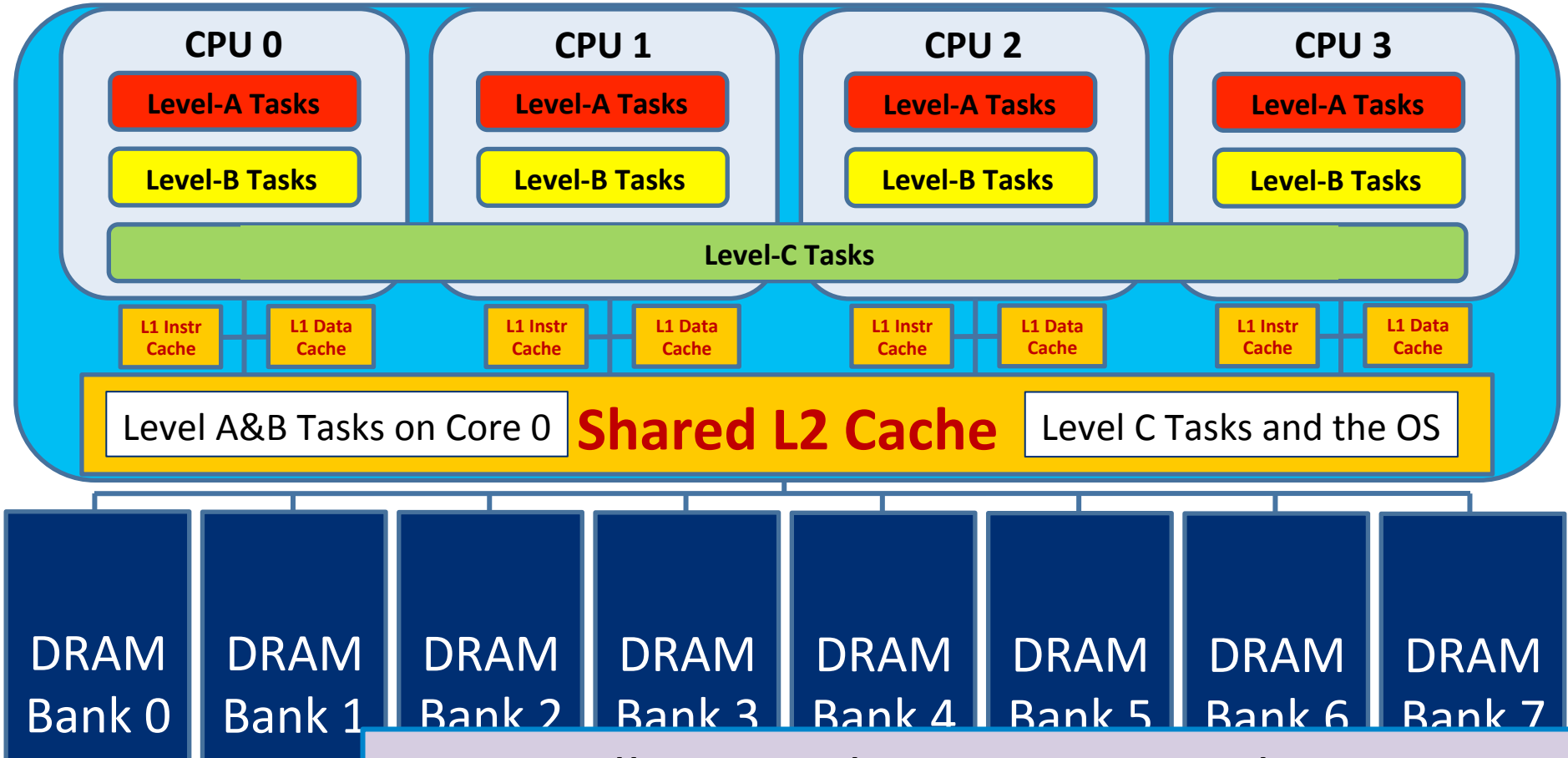
MC² (Mixed-Criticality on Multicore)



We can allocate arbitrary rectangular regions of the **shared L2 cache** to sets of tasks.

CPS 1239135, PI: James H. Anderso
UNC Chapel Hill, co-PI: Frank Mueller,
NC State University.

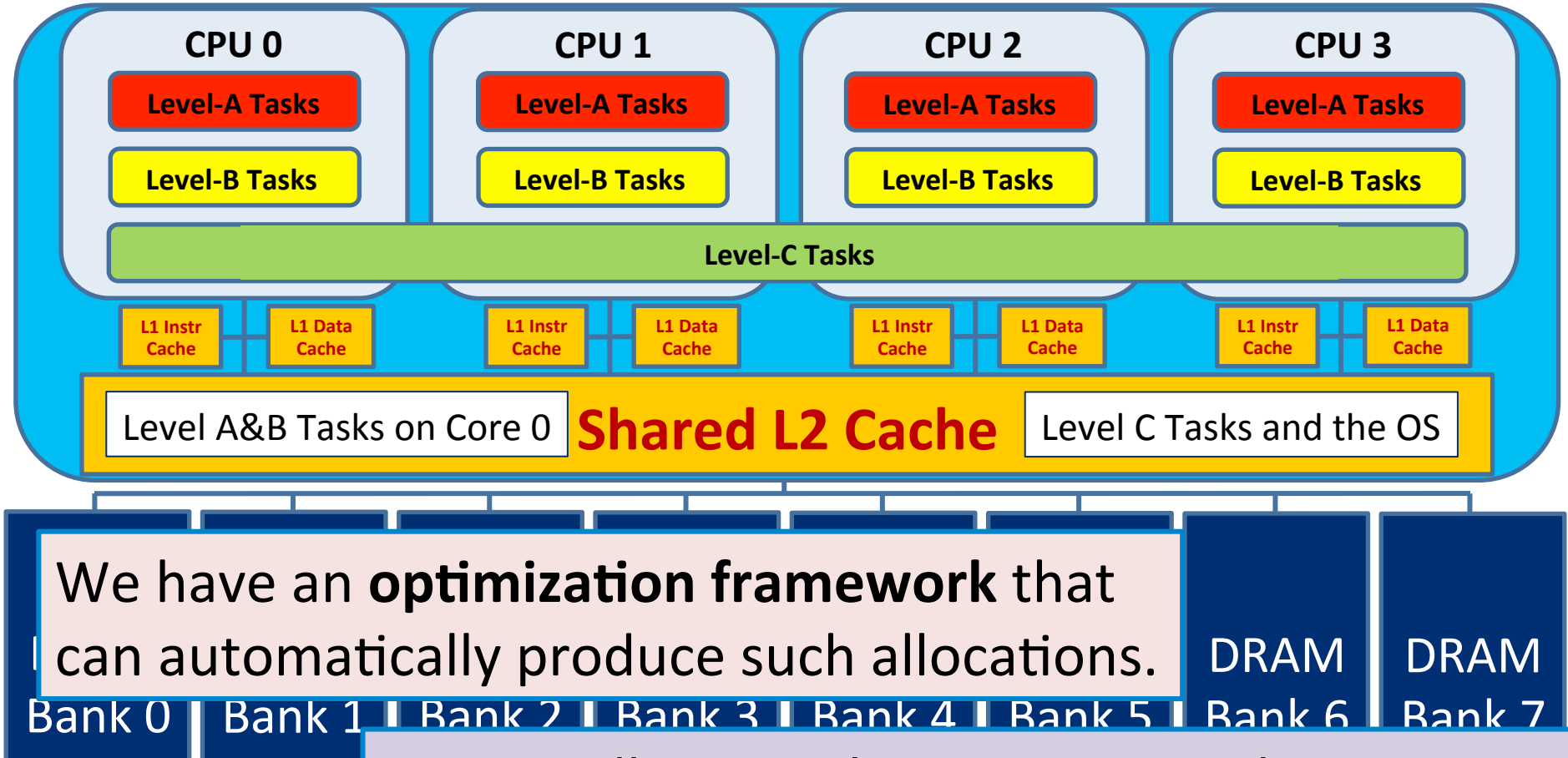
MC² (Mixed-Criticality on Multicore)



We can allocate arbitrary rectangular regions of the **shared L2 cache** to sets of tasks.

CPS 1239135, PI: James H. Anderso
UNC Chapel Hill, co-PI: Frank Mueller,
NC State University.

MC² (Mixed-Criticality on Multicore)

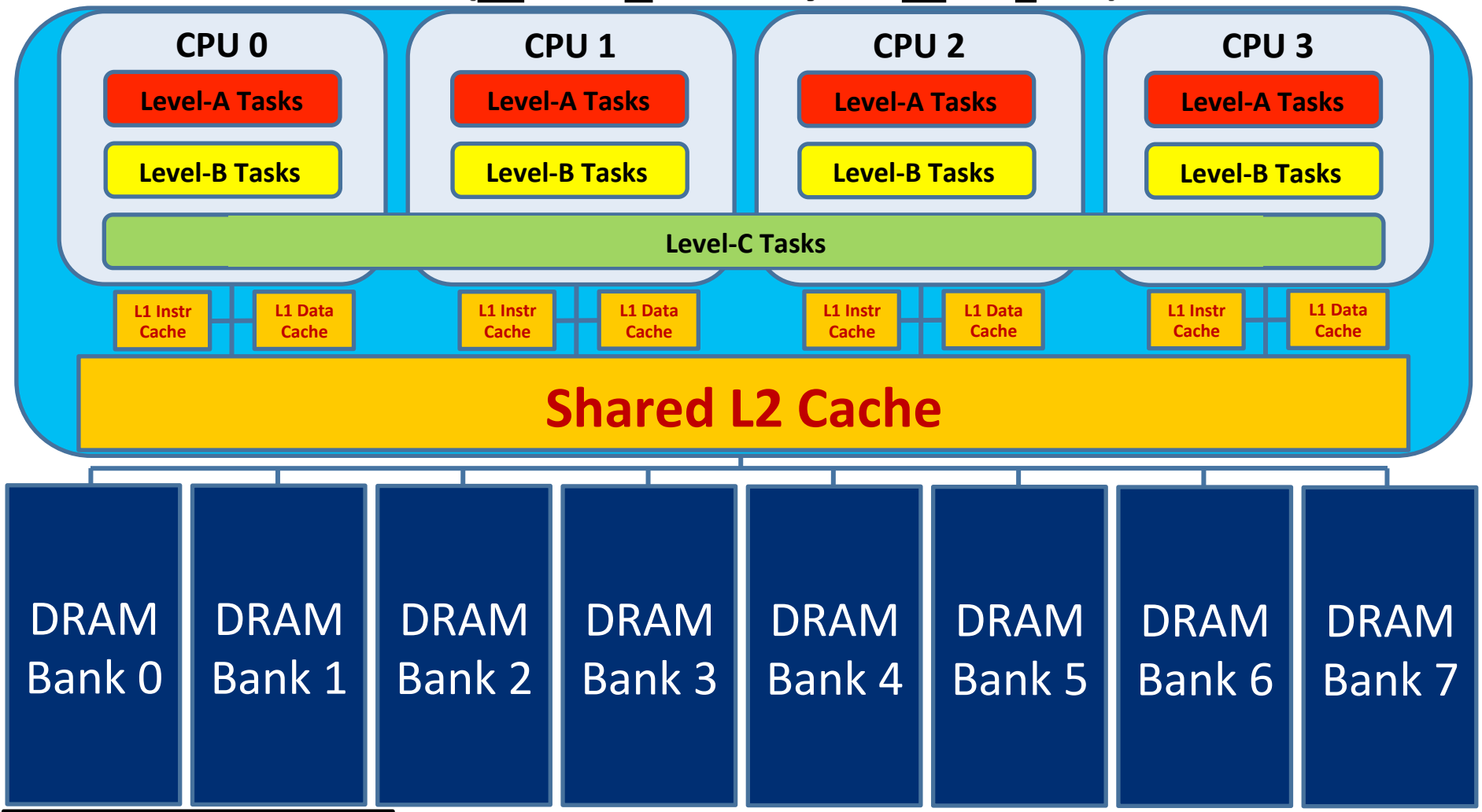


We have an **optimization framework** that can automatically produce such allocations.

We can allocate arbitrary rectangular regions of the **shared L2 cache** to sets of tasks.

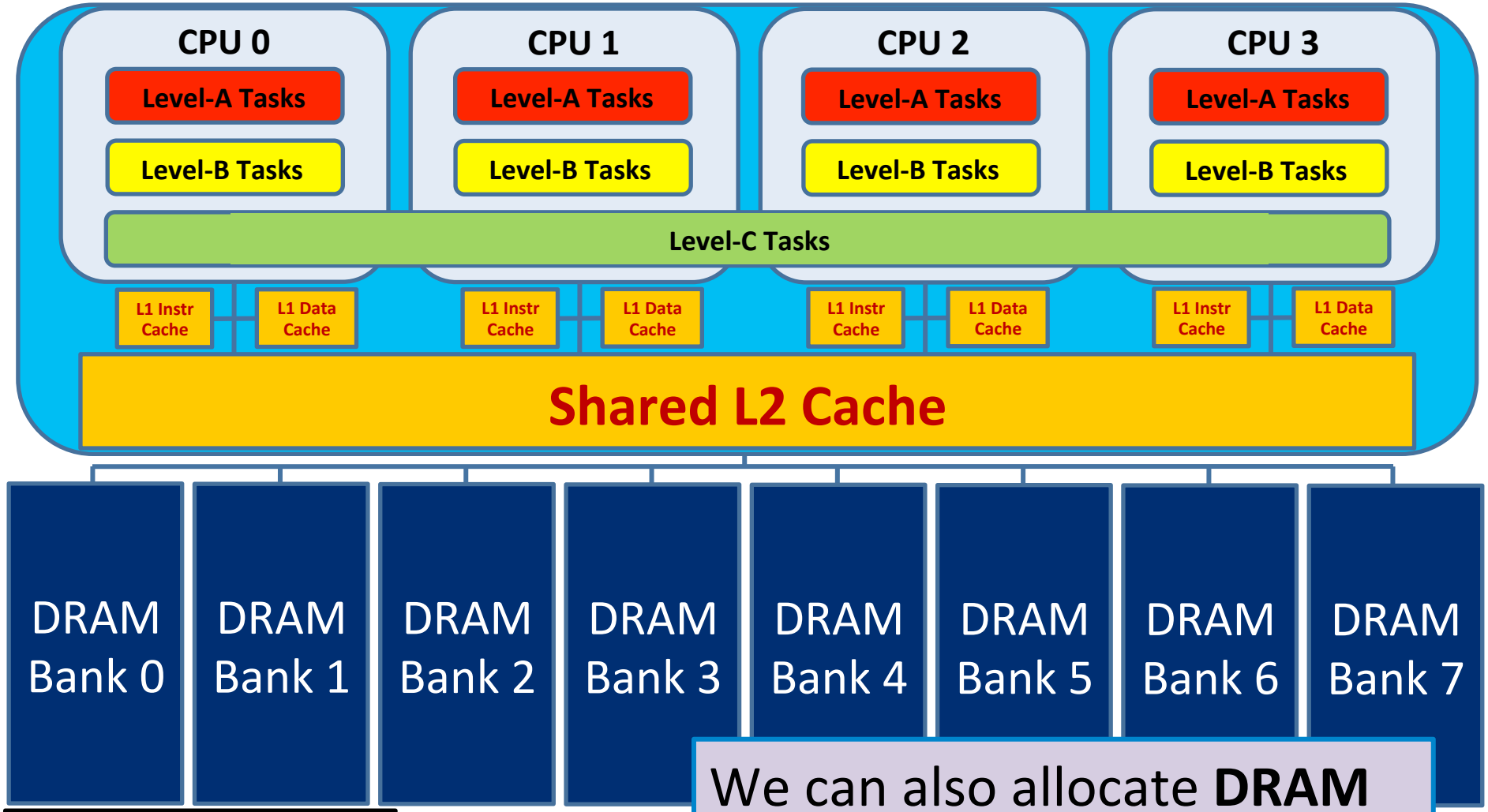
CPS 1239135, PI: James H. Anderso
UNC Chapel Hill, co-PI: Frank Mueller,
NC State University.

MC² (Mixed-Criticality on Multicore)



CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

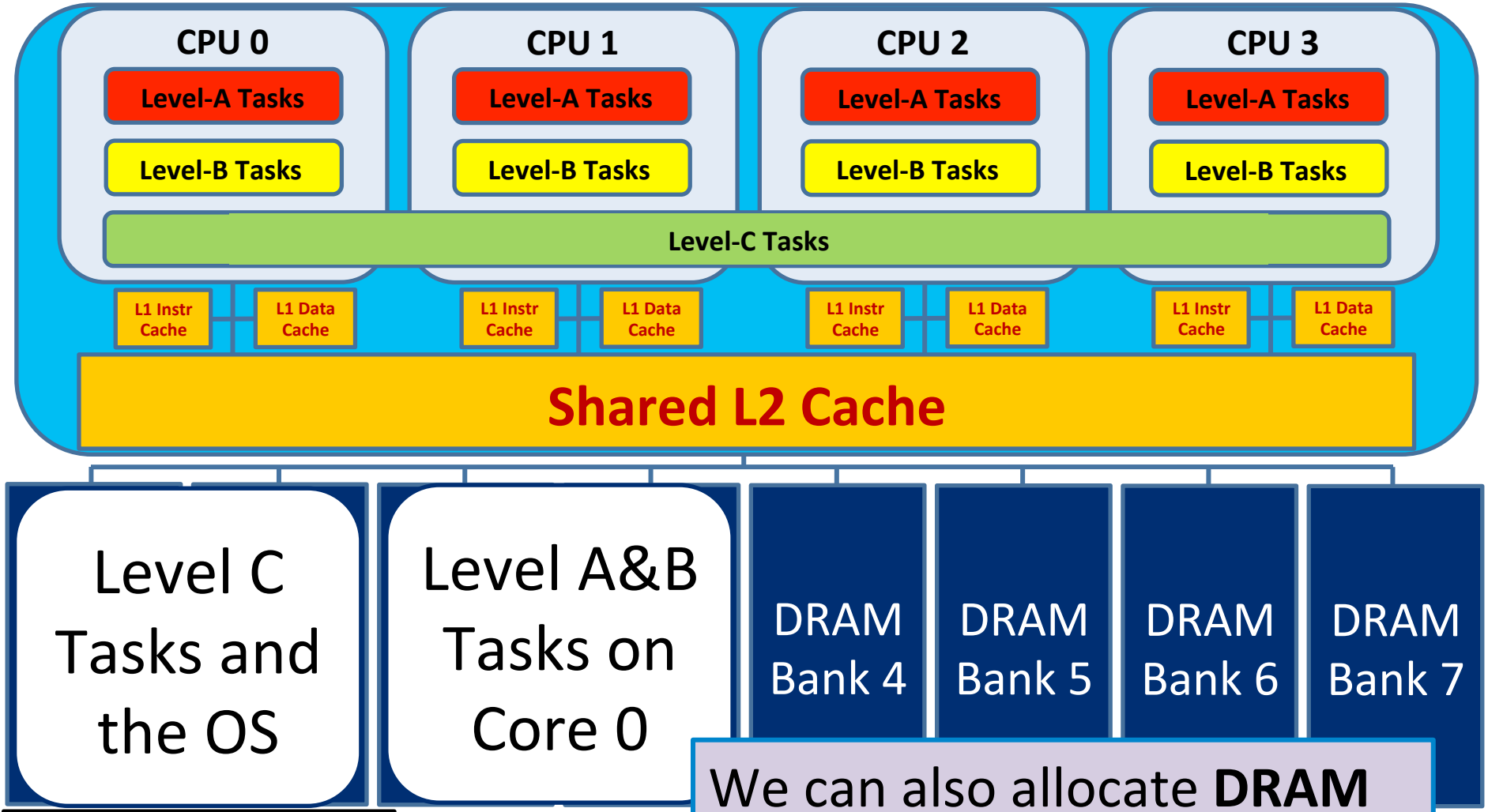
MC² (Mixed-Criticality on Multicore)



We can also allocate **DRAM banks** to certain sets of tasks.

CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

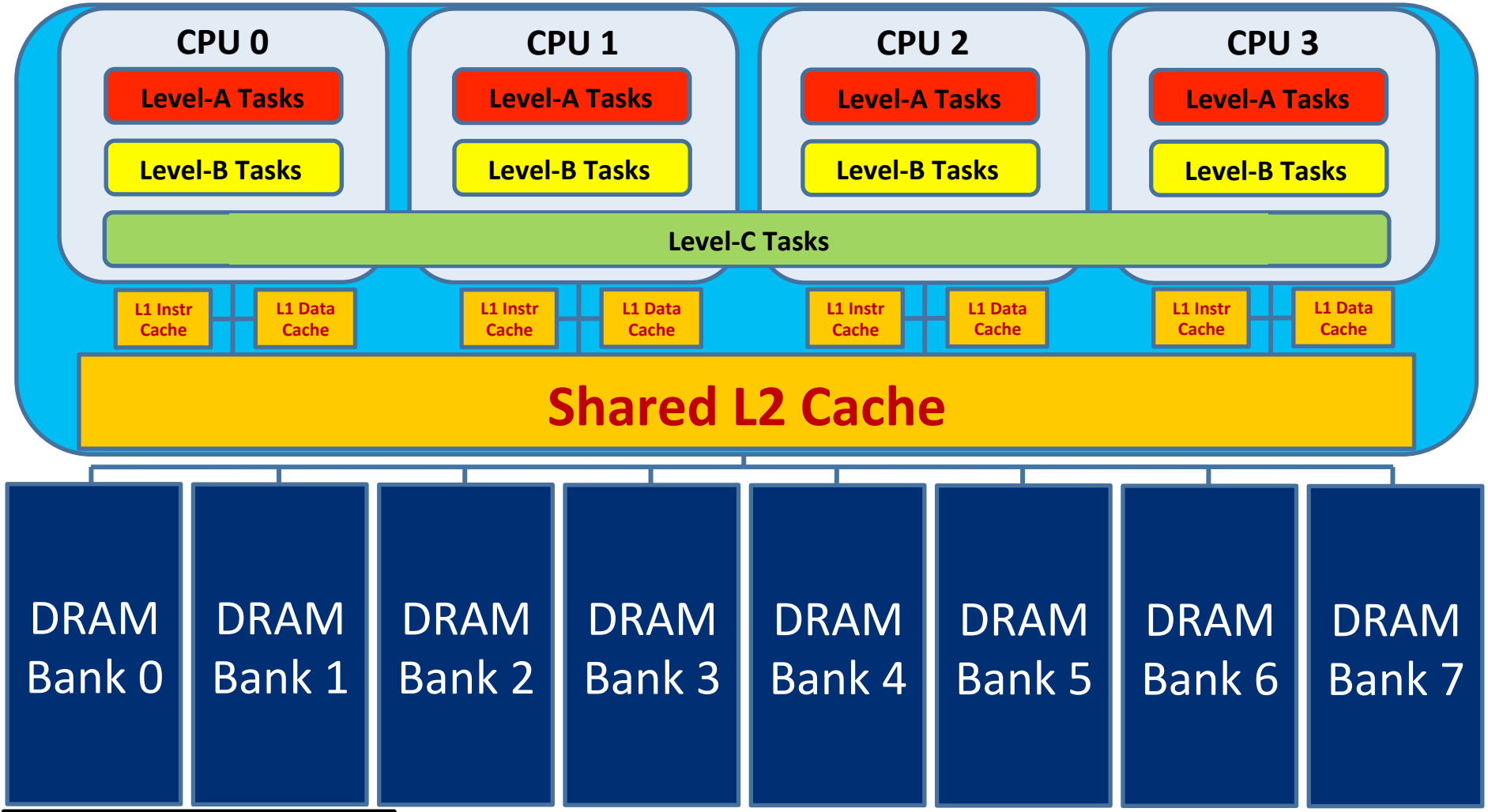
MC² (Mixed-Criticality on Multicore)



We can also allocate **DRAM banks** to certain sets of tasks.

CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

MC² (Mixed-Criticality on Multicore)



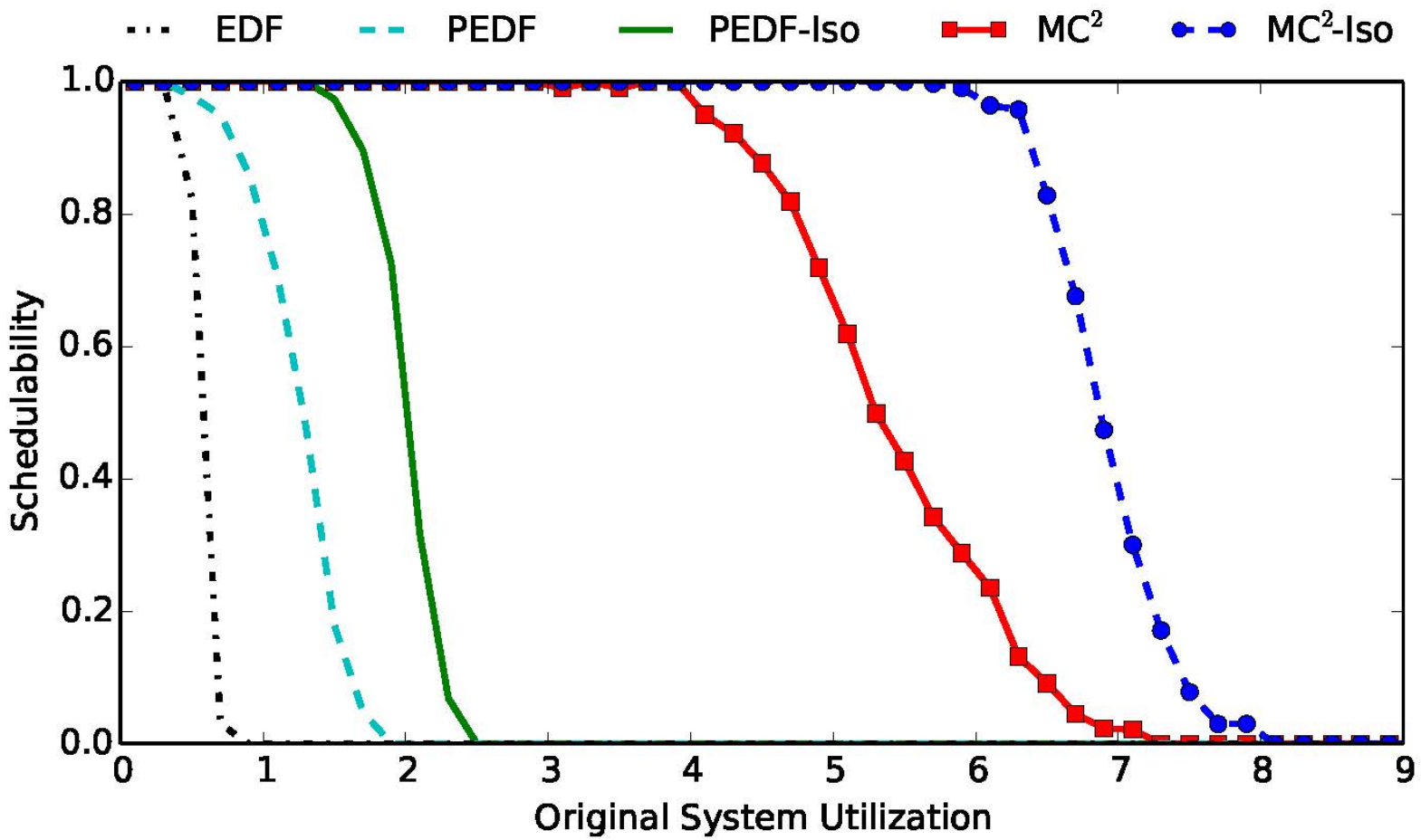
CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

Outline

- Real-time 101.
- Problems caused by multicore.
  - "The one-out-of-m problem."
  - Why this is an important problem.
- Solution strategy.
- Evaluation.
- Project summary.

CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

A Little Experimental Evidence in Favor of our Approach

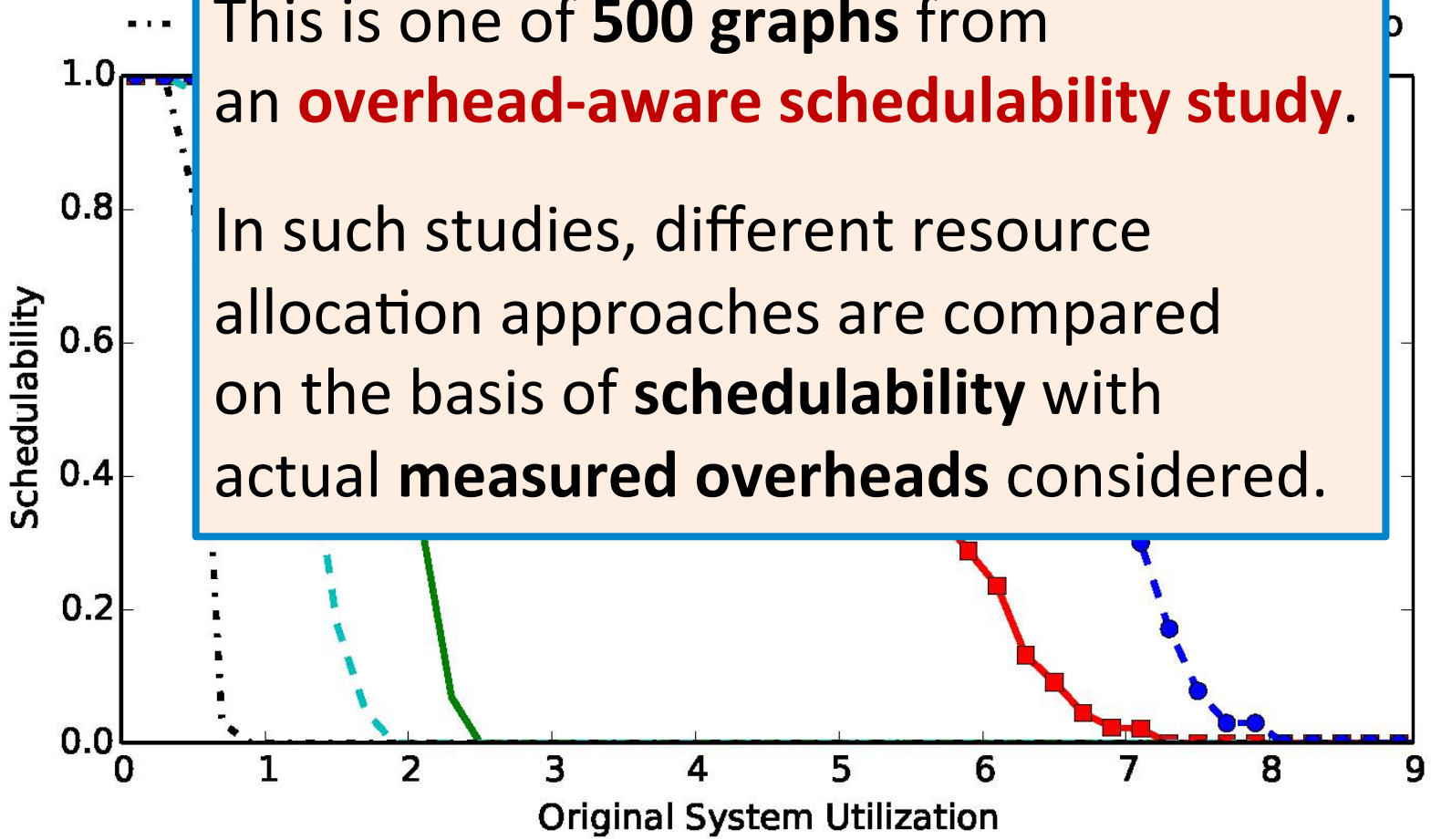


CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

A Little Experimental Evidence in Favor of our Approach

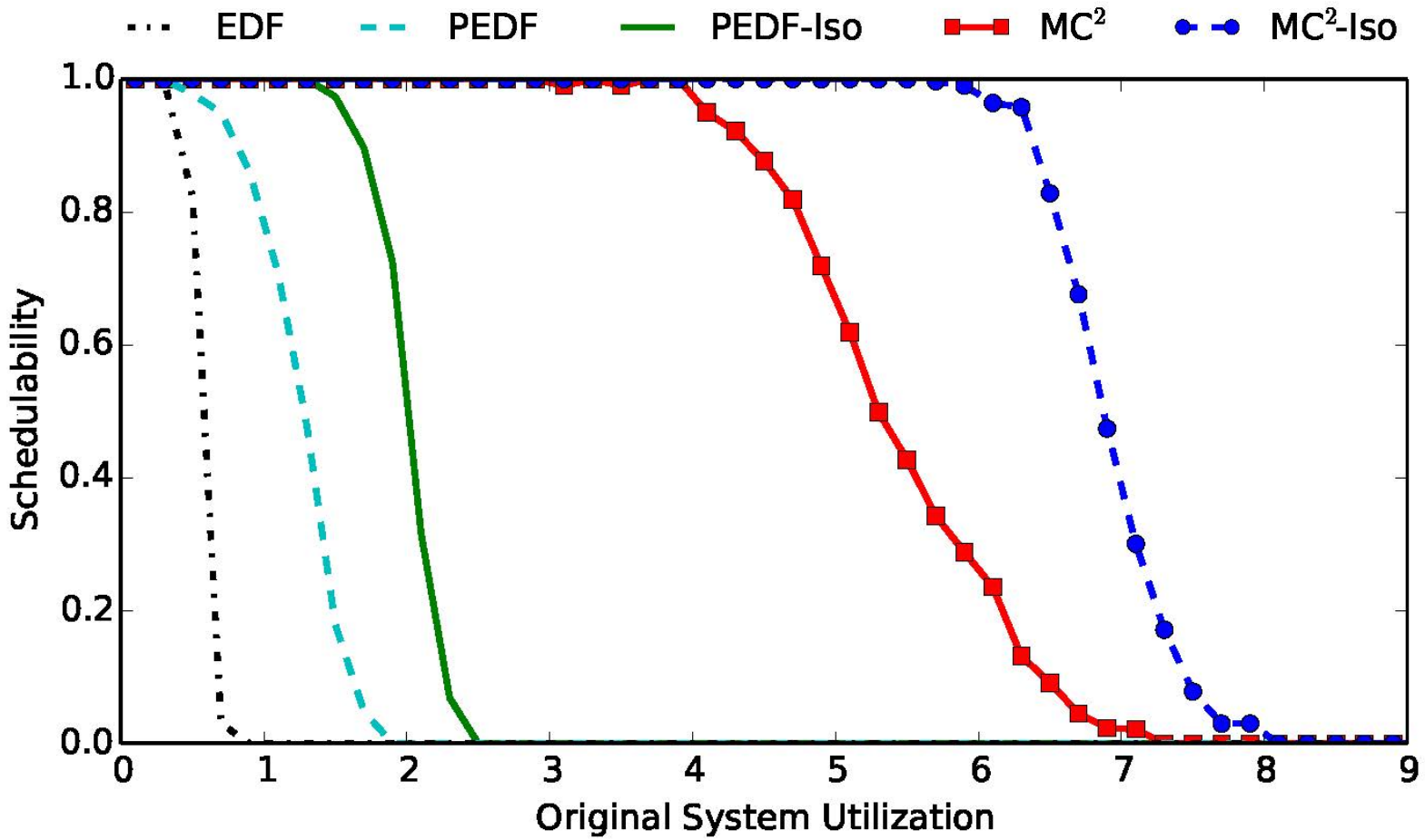
This is one of **500 graphs** from an **overhead-aware schedulability study**.

In such studies, different resource allocation approaches are compared on the basis of **schedulability** with actual **measured overheads** considered.



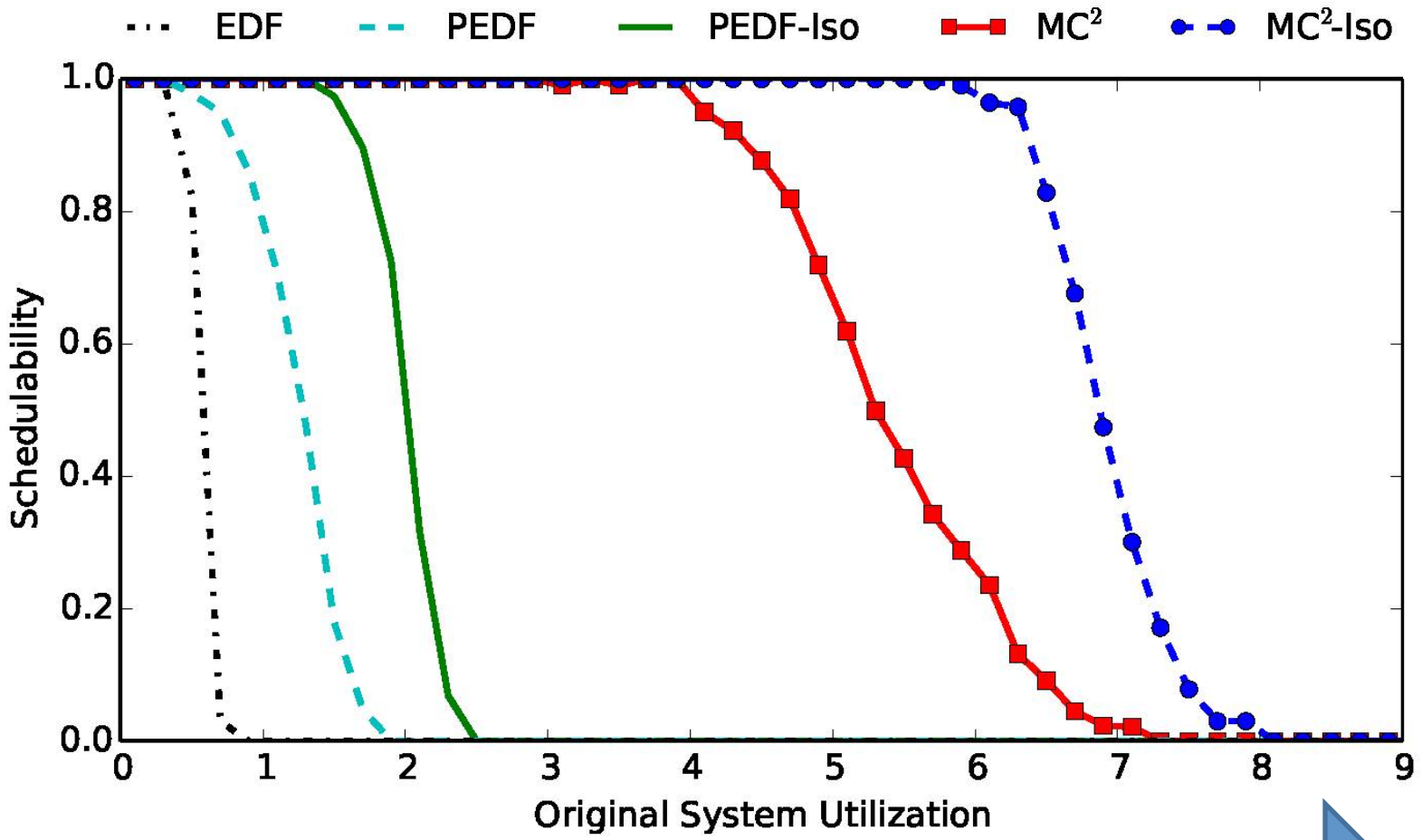
CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

A Little Experimental Evidence in Favor of our Approach



CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

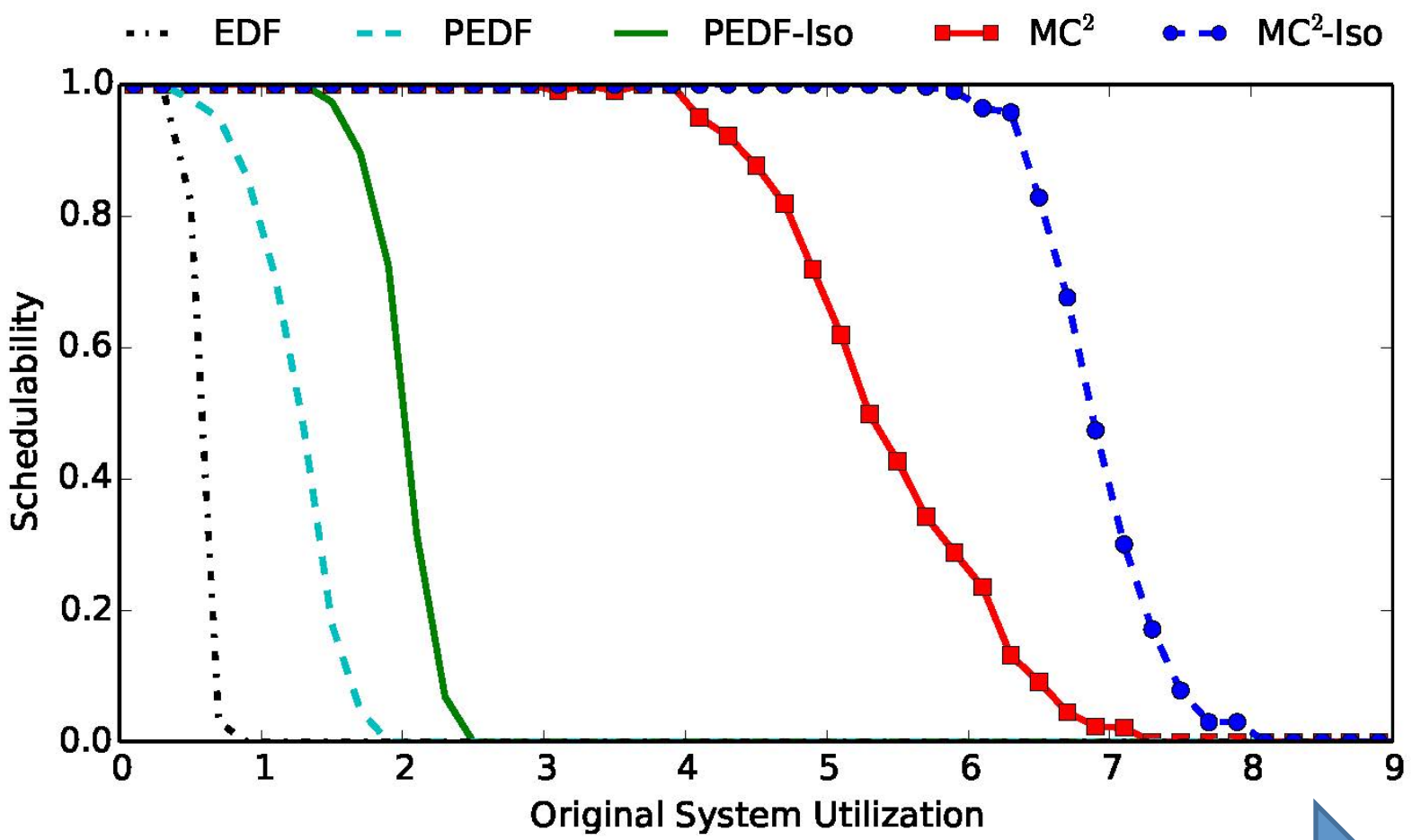
A Little Experimental Evidence in Favor of our Approach



Total System Utilization

CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

A Little Experimental Evidence in Favor of our Approach

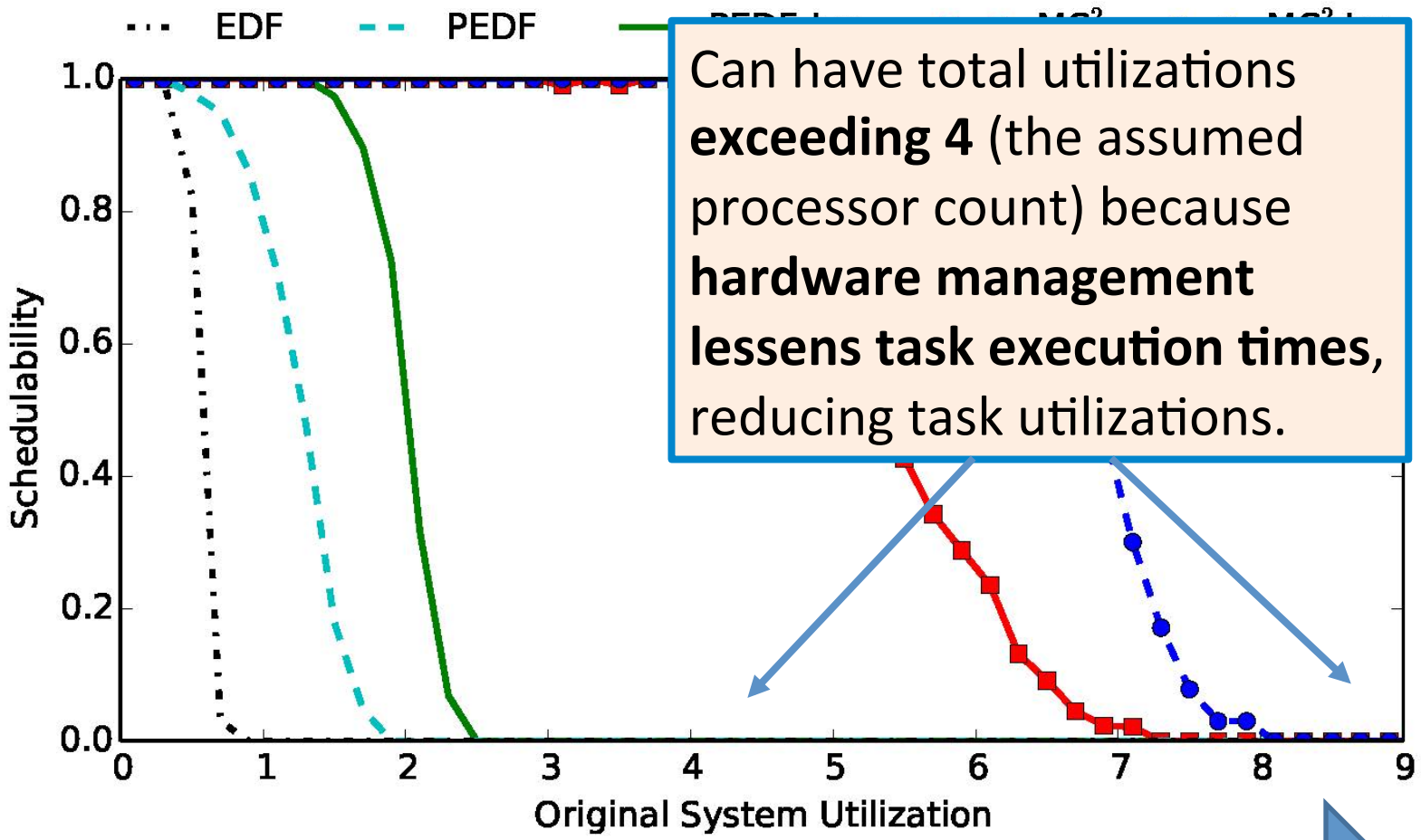


Fraction of Task Systems Deemed Schedulable

Total System Utilization

CPS 12 : James H. Anderson, UNC Cl , co-PI: Frank Mueller, NC Sta sity.

A Little Experimental Evidence in Favor of our Approach



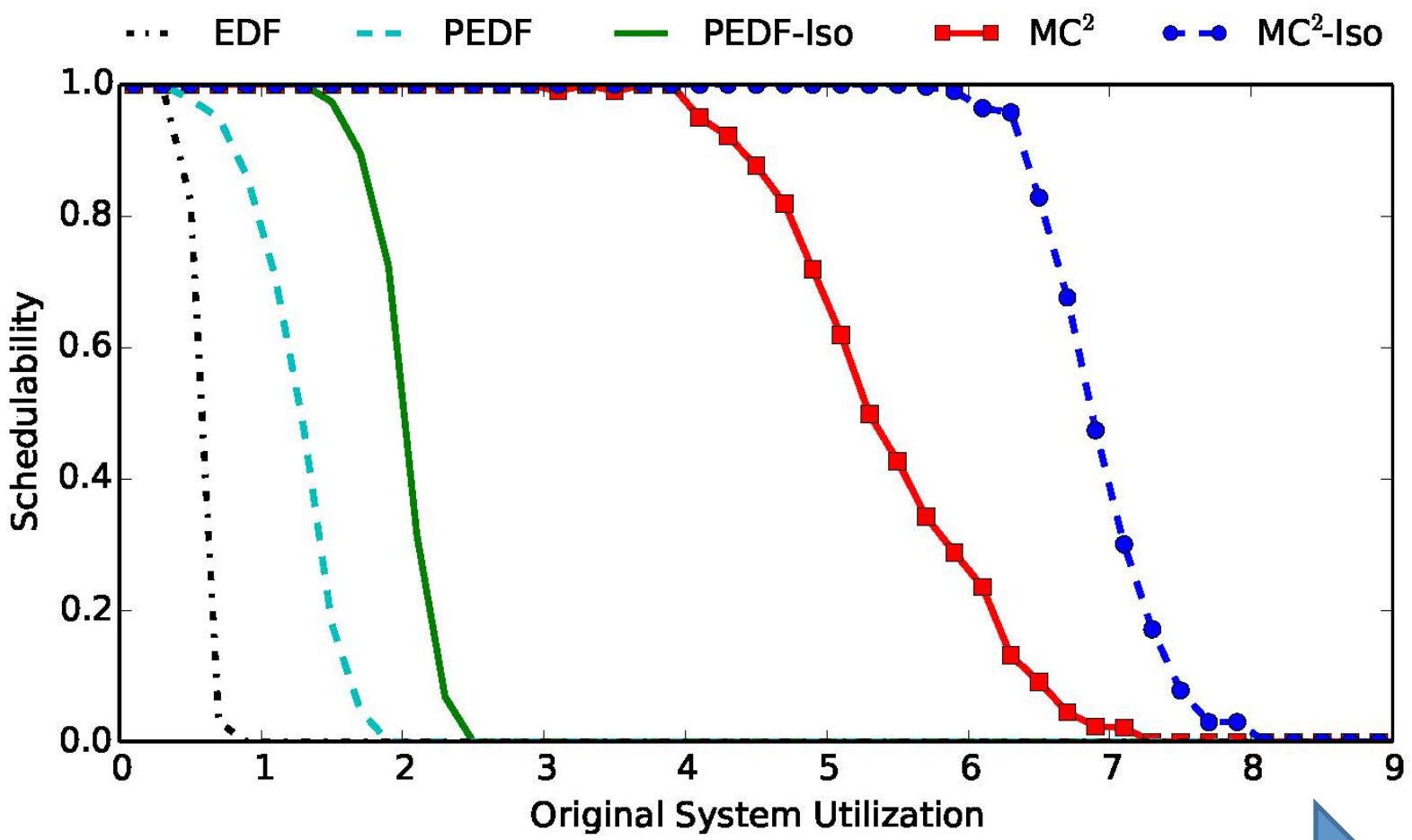
Can have total utilizations **exceeding 4** (the assumed processor count) because **hardware management lessens task execution times**, reducing task utilizations.

Fraction of Task Systems Deemed Schedulable

Total System Utilization

CPS 12 : James H. Anderson, UNC Cl , co-PI: Frank Mueller, NC Sta sity.

A Little Experimental Evidence in Favor of our Approach

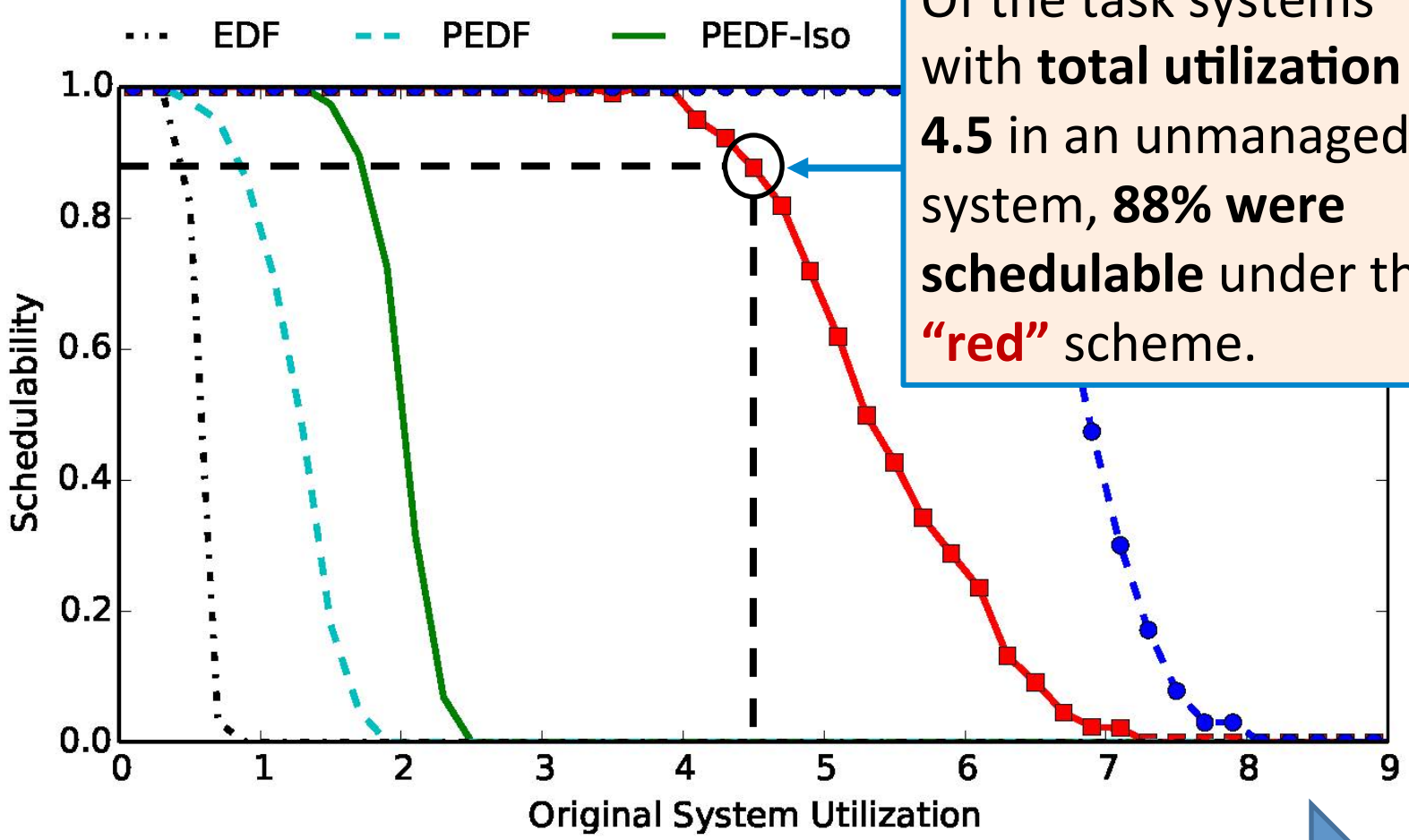


Fraction of Task Systems Deemed Schedulable

Total System Utilization

CPS 12 : James H. Anderson, UNC Cl , co-PI: Frank Mueller, NC Sta sity.

A Little Experimental Evidence in Favor of our Approach



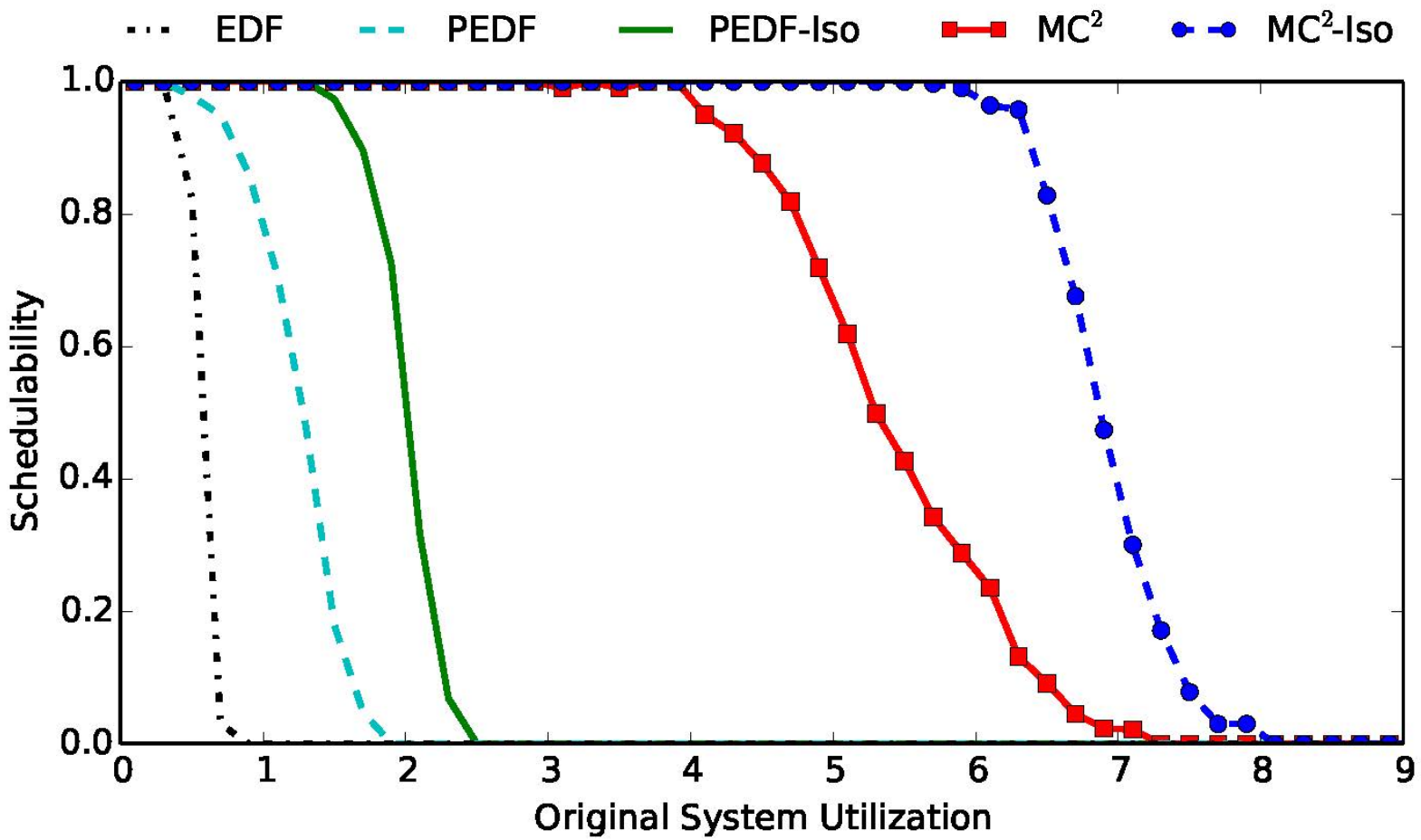
Of the task systems with **total utilization 4.5** in an unmanaged system, **88% were schedulable** under the **"red"** scheme.

Fraction of Task Systems Deemed Schedulable

Total System Utilization

CPS 12 : James H. Anderson, UNC Cl , co-PI: Frank Mueller, NC Sta sity.

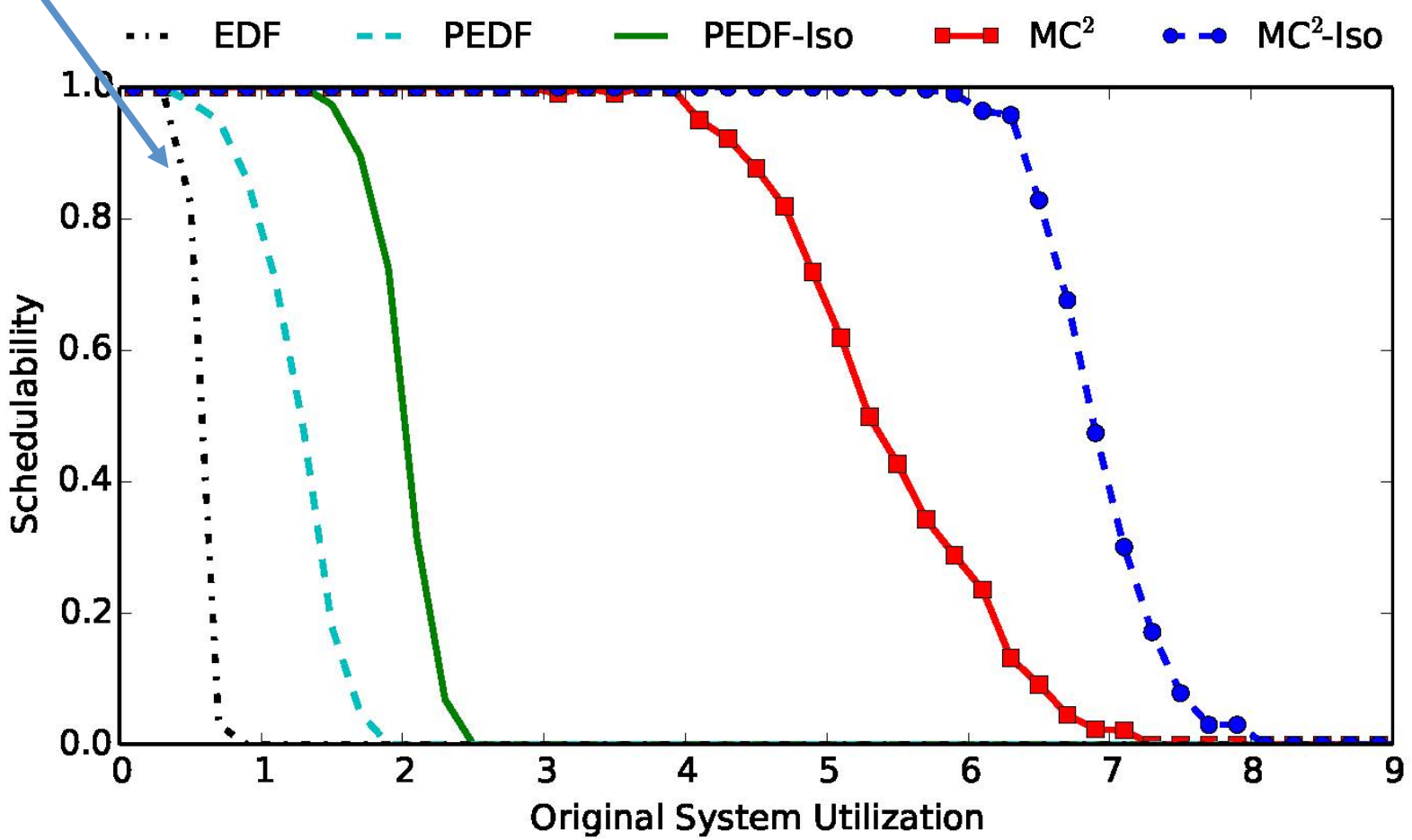
A Little Experimental Evidence in Favor of our Approach



CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

Uniprocessor EDF (the current de facto standard)

rimental Evidence in Favor of our Approach

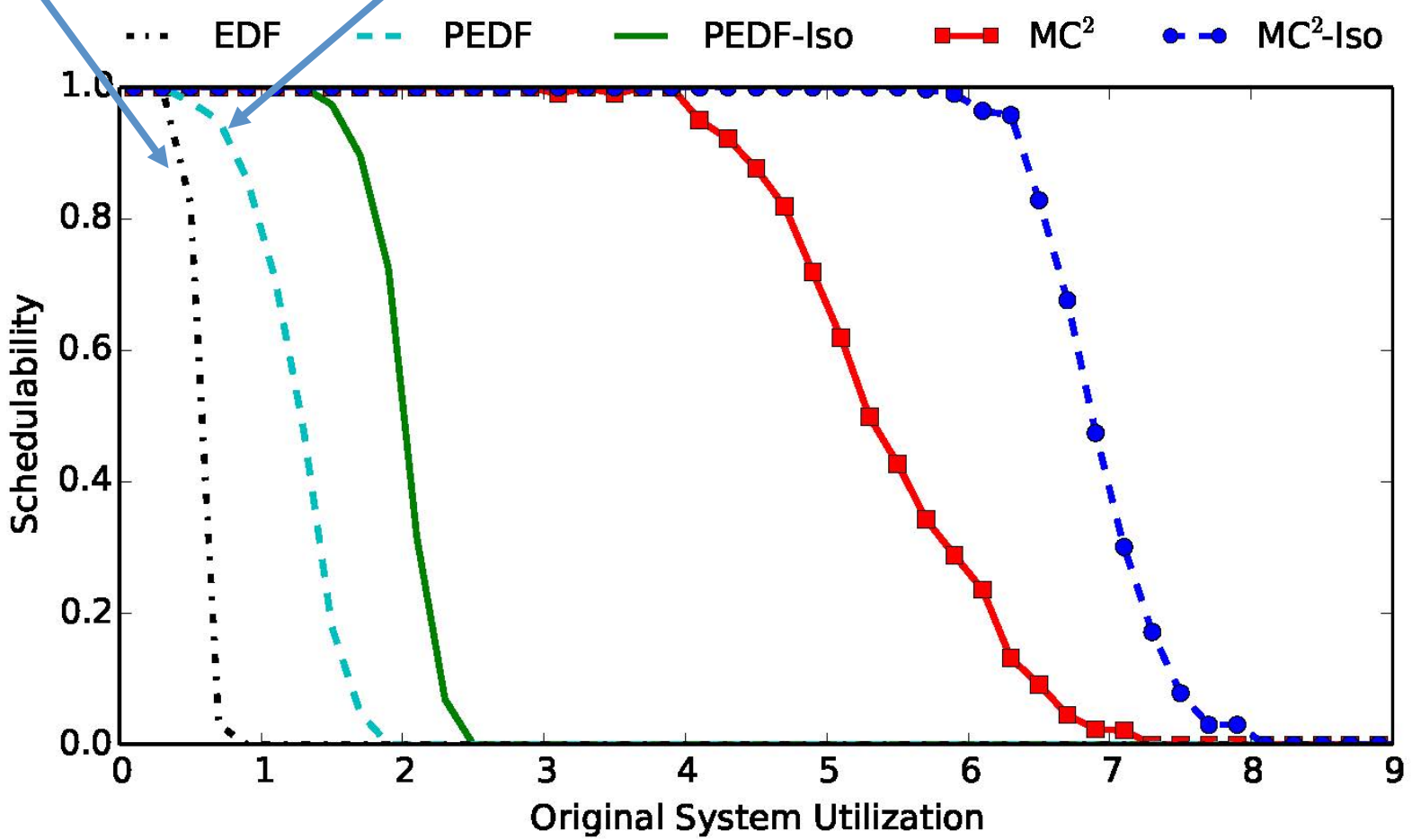


CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

Uniprocessor EDF (the current de facto standard)

Partitioned EDF

rimental Evidence in Favor of our Approach



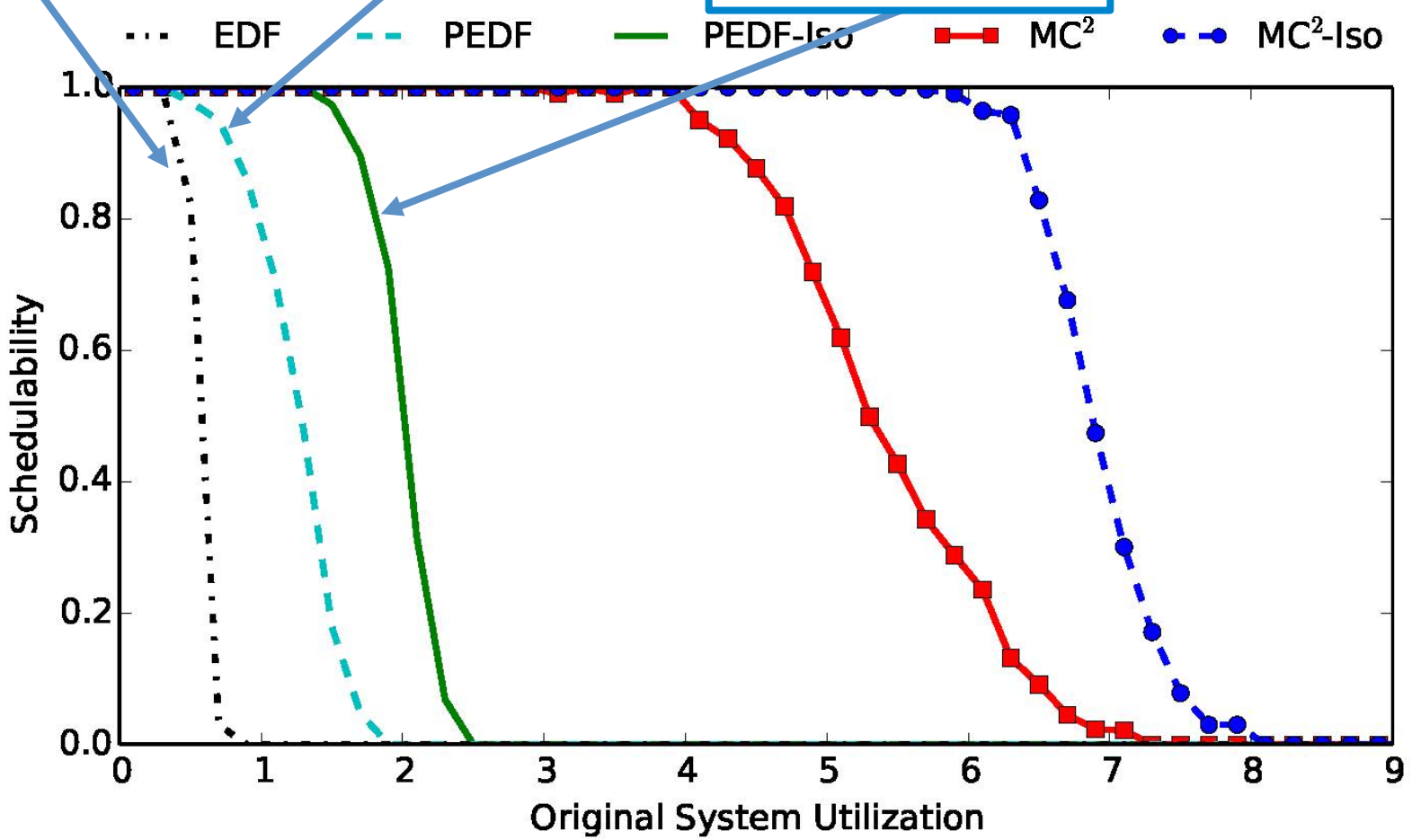
CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

Uniprocessor EDF (the current de facto standard)

Partitioned EDF

Partitioned EDF with hardware management

rimental Evidence pproach



CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

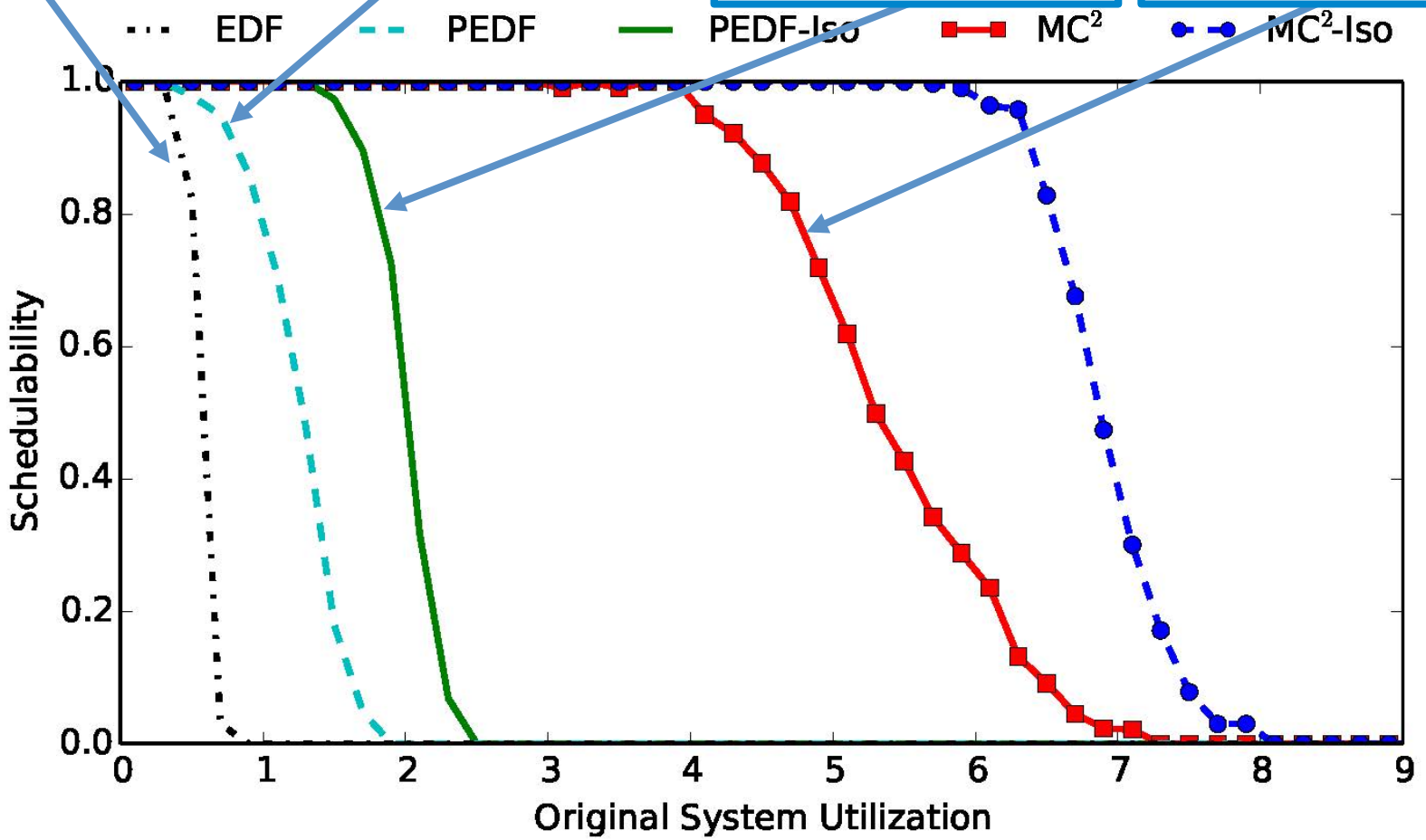
Uniprocessor EDF (the current de facto standard)

Partitioned EDF

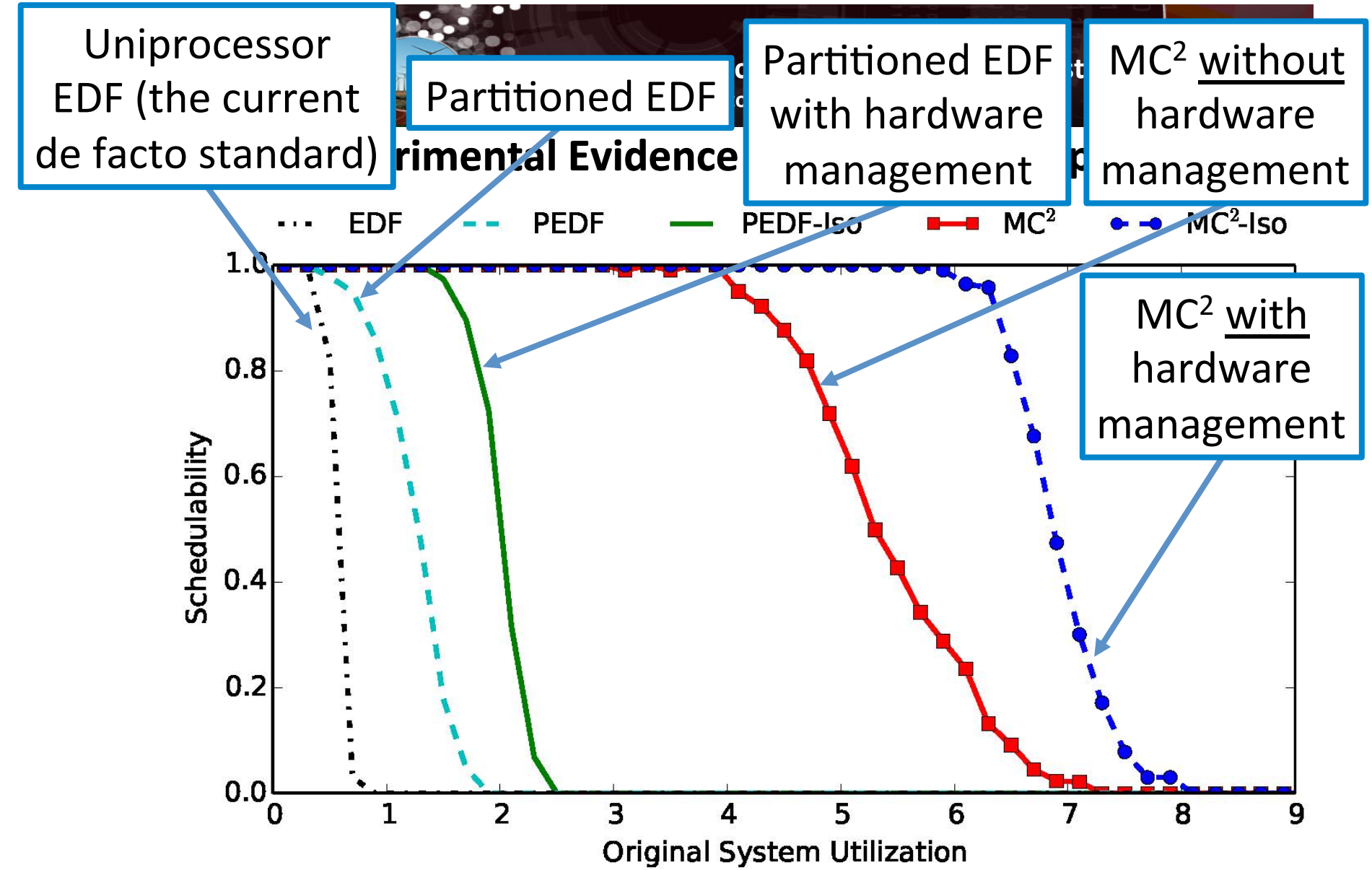
Partitioned EDF with hardware management

MC² without hardware management

rimental Evidence

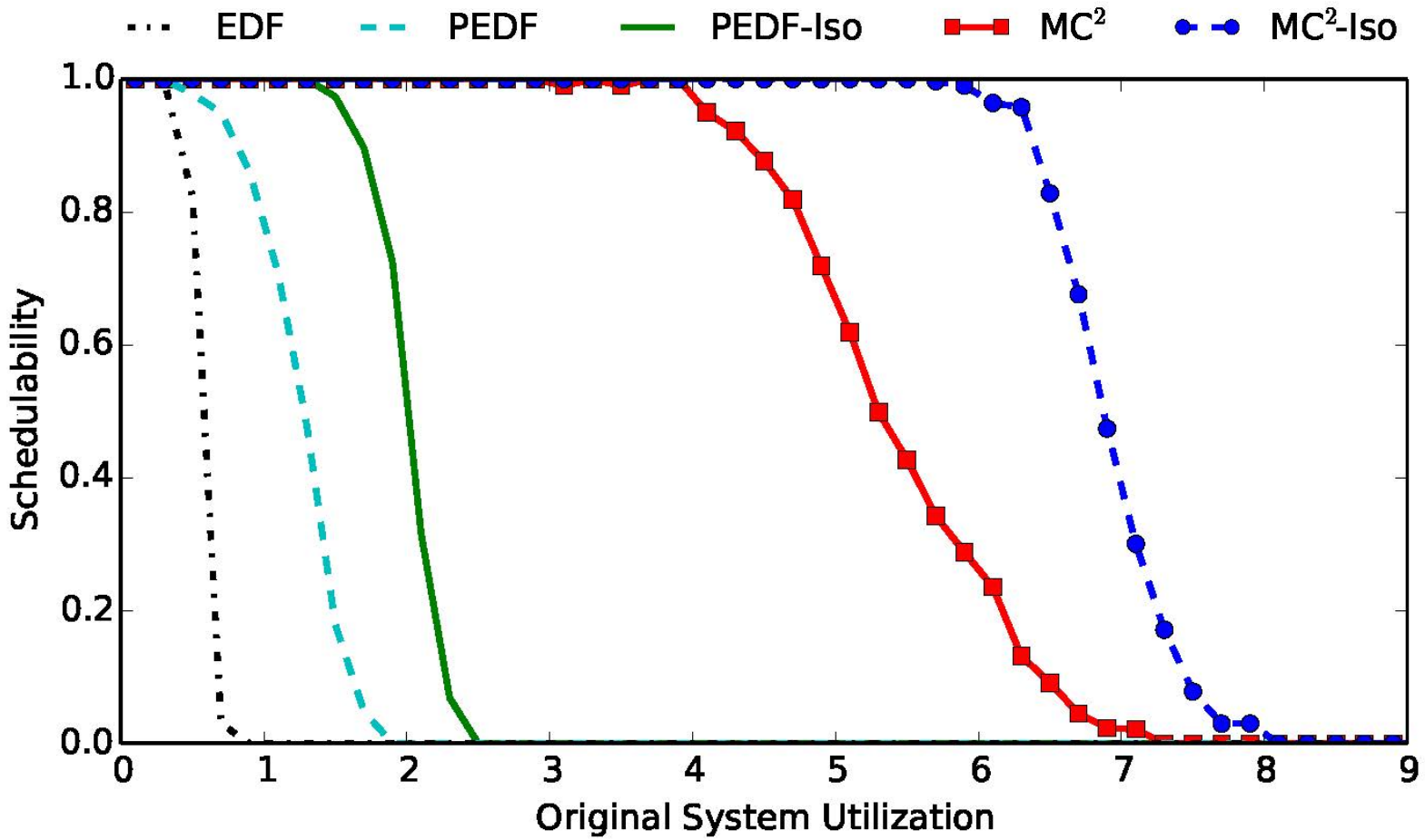


CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.



CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

A Little Experimental Evidence in Favor of our Approach



CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.

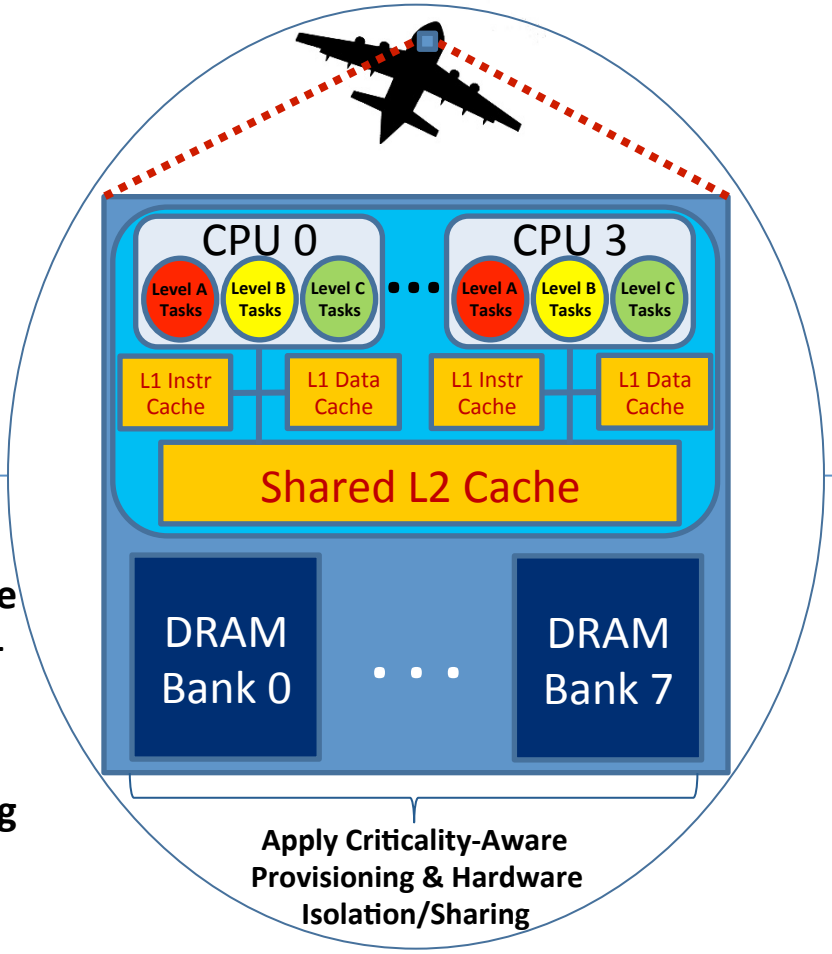
Project Summary
CPS: Breakthrough: Collaborative Research: Bringing the Multicore Revolution to Safety-Critical Cyber-Physical Systems

Challenge:

- To enable the usage of **multicore platforms** in safety critical domains, particularly avionics.
- On multicore, **cross-core interactions** are hard to predict.

Solution:

- Combine **shared hardware management** with **mixed-criticality provisioning**.
- Enforce **isolation** where needed but enable **sharing** where possible.



Scientific Impact:

- Proposed framework can be generally applied to support **computationally intensive real-time workloads** on **multicore**.
- Workloads **4-8× "larger"** were enabled in experiments.

Broader Impact:

- Adopted approach devised in consultation with **avionics industry** colleagues.
- Research directly targets concerns raised in the FAA's **CAST 32** report.
- Two new **CPS courses** created.

CPS 1239135, PI: James H. Anderson, UNC Chapel Hill, co-PI: Frank Mueller, NC State University.