Single Core Equivalence Framework (SCE) For Certifiable Multicore Avionics

http://rtsl-edge.cs.illinois.edu/SCE/

a collaboration of:

Presenters: Lui Sha, Marco Caccamo, Heechul Yun, Renato Mancuso, Jung-Eun Kim

With contributions from
Rodolfo Pellizzoni and Man Ki Yoon
Russell Kegley, Jonathan Preston and Dennis Perlman
Greg Arundale and Richard Bradford

Sponsored in part by:

NSERC CRSNG LOCKHEED MARTIN Rockwell Collins

Outline

SCE

single-core equivalence

1. Overview
2. CAST32 and SCE
3. DRAM, Bus and Cache management
4. IMA and I/O management
5. SCE Summary

Transition to Multi-Core

multi-core benefits:

- reduced **space** and **weight**
- reduced **power** and **cooling**
- increased **computation**
- … and more

We have a **large body** of certified **single-core** hard real-time **software**

Existing software will be **migrated en-masse**, in addition to new software.

There are **serious risks**

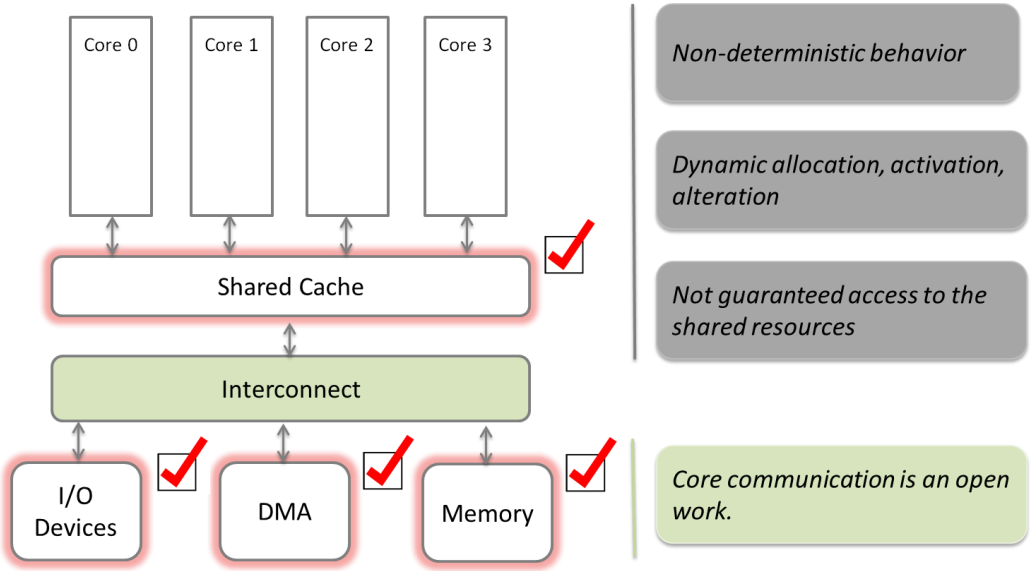
Inter-core Interferences

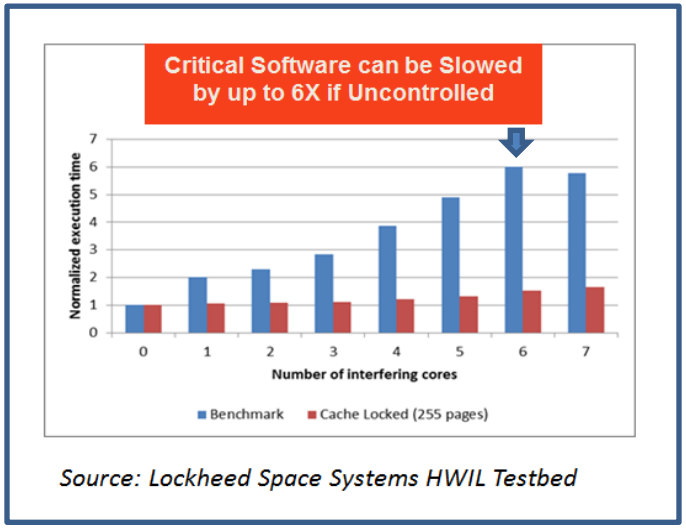
- *"Shared access to cache or other memory areas, operating systems / supervisors / hypervisors that can control and affect all the applications executing on all the cores, and 'coherency fabrics / coherency modules / interconnects' that control all the data transfers between the MCP cores, memory and the peripheral devices of the MCP via a shared bus.*

- ***Many of these features … were not designed or verified for compliance with the current airborne software or hardware guidance material.***

- *It may therefore be difficult or even impossible to fully characterize and verify all the possible effects of these features, which may include unintended and unexpected behavior."*

- CAST 32

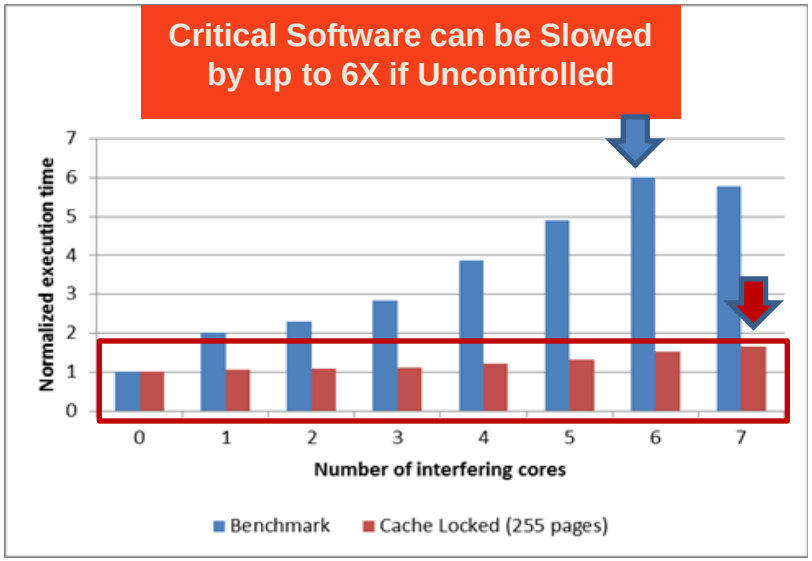




Source: Lockheed Space Systems HWIL Testbed

Testbed Results

- Lockheed Space Systems ported some applications to a Freescale P4080 testbed. Tests indicated that
- Recorded max delay (**blue bar**) of a task increased 6x, when **7** cores were used, *but not when **8** cores were used.* This makes the determination of worst case configuration difficult.
- Using SCE technology (**red bar**)
  - Recorded max delay of a task increased monotonically when more cores were used.
  - WCET(**8**) > WCET(j), j = 1 to 7; and increased less than 2x.



Source: Lockheed Space Systems HWIL Testbed

- Currently SCE core **isolation technology is software based**. The isolation overhead increases when more cores are used.
- **Hardware design** support can greatly **reduce the overhead** but requires cooperation from chip makers.

DO178C and the WCET Assumption

- **DO178 C was developed for single core chips** under the assumption that a acceptably tight bound on a task T's execution time (WCET) can be determined and reused for different task sets in a given platform.
- This **constant WCET assumption** makes schedulability analysis, timing tests and timing certification tractable. Without it, any change to any task mandates the recalculation of all other tasks' WCETs.
- In a multicore chip, as is, physically concurrent sharing of globally available DRAM banks, memory bus, last level cache, and I/O channels invalidates the traditional constant WCET assumption.
- SCE generalizes the constant WCET assumption in the form of constant WCET(m) assumption, where m is the maximal number of cores that will be used in a multicore chip. **SCE allows the reuse of DO178 C as is with WCET(1), a.k.a, WCET, replaced by WCET(m).**

Single Core Equivalence Framework (SCE) For Certifiable Multicore Avionics

SCE and CAST-32

a collaboration of:

Presenter: Marco Caccamo

Sponsored in part by:

NSERC CRSNG

LOCKHEED MARTIN

Rockwell Collins

CAST-32 Position

CAST-32/position

- **DO-178B** and **DO-178C** only address software on **single-core processor**
- In MCPs, applications on **separate cores** may cause i**nterference** with each other
- **No existing material** to adapt development and verification on MCPs

1	MCP Interference Channels	*position d*
2	Shared Memory and Cache	*position e*
3	Planning and Verification of Resource Usage	*position f*
4	Software Verification	*position h*

SCE Overview

SCE

single-core equivalence

is

a framework of **OS-level** techniques

implementable on **commercial MCP** platforms

for **strict partitioning** of **shared resources**

so that **each core** can be treated as a **single-core** chip

from a **schedulability analysis**

and **certification perspective**

CAST-32 Coverage

SCE

single-core equivalence

and

CAST-32

1

MCP Interference Channels

*"Applications running on different cores of a MCP **do not execute independently** from each other because the cores are **sharing resources**"*

CAST-32/position d.i

*"The applicant has conducted a functional **interference analysis** […] and has designed, implemented and verified a **means of mitigation** for each interference channel"*

CAST-32/position d.ii

Within **SCE** we have:

- **Identified and analyzed** the main interference channels
- Provided a **mitigation strategy** for each channel
- Exported a set of **equivalent, independent single-cores**

CAST-32 Coverage

SCE

single-core equivalence

and

CAST-32

2

Shared Memory and Cache

*"WCET of the software applications hosted on one core **can increase greatly** due to **repeated cache accesses** by the **processes hosted on the other core**"*

CAST-32/position e.i

"The applicants have to describe their strategy for managing and verifying cache usage" and *"to conduct analyses of worst-case effect of shared cache"*

CAST-32/position e.ii

SCE provides:

- Per-process cache usage **profiling mechanism**
- **Deterministic** shared cache **allocation strategy**
- **No inter-** and **intra-core interference** on cache space

CAST-32 Coverage

SCE

single-core equivalence

and

CAST-32

3

Planning and Verification of Resource Usage

*"If the **overall available resources** of the MCP are **exceeded** by the combined resource demand, the effects on the software **may be unpredictable**"*

CAST-32/position f.i

*"The applicants have to describe their plans to allocate, manage and measure **the use of the interconnect** used by applications and peripherals"*

CAST-32/position f.ii

SCE provides:

- Per-core **memory bandwidth regulation** mechanism
- **Guarantee** of operation below **saturation point**
- **Serialization** of **I/O transactions**

CAST-32 Coverage

CMAS

SCE

single-core equivalence

and

CAST-32

4

Software Verification

*"**Existing guidance** and **standard industry practice** for the integration and verification of hardware platforms, OSes and applications is the field of **IMA systems**"*

CAST-32/position h.i

*"A **similar approach** […] would be **effective** to the verification of software on an MCP"* since it *"would not impose any **additional burden** on the industry"*

CAST-32/position h.i

In summary**, using SCE**:

- Perform per-core **modular analysis** and **certification**
- Reuse **consolidated software** and **engineering processes**
- Use an **IMA approach** on each equivalent single-core
- Verification of SCE implementation is an open challenge

Single Core Equivalence Framework (SCE) For Certifiable Multicore Avionics

Tech. Overview and Cache Management

a collaboration of:

Presenter: Renato Mancuso

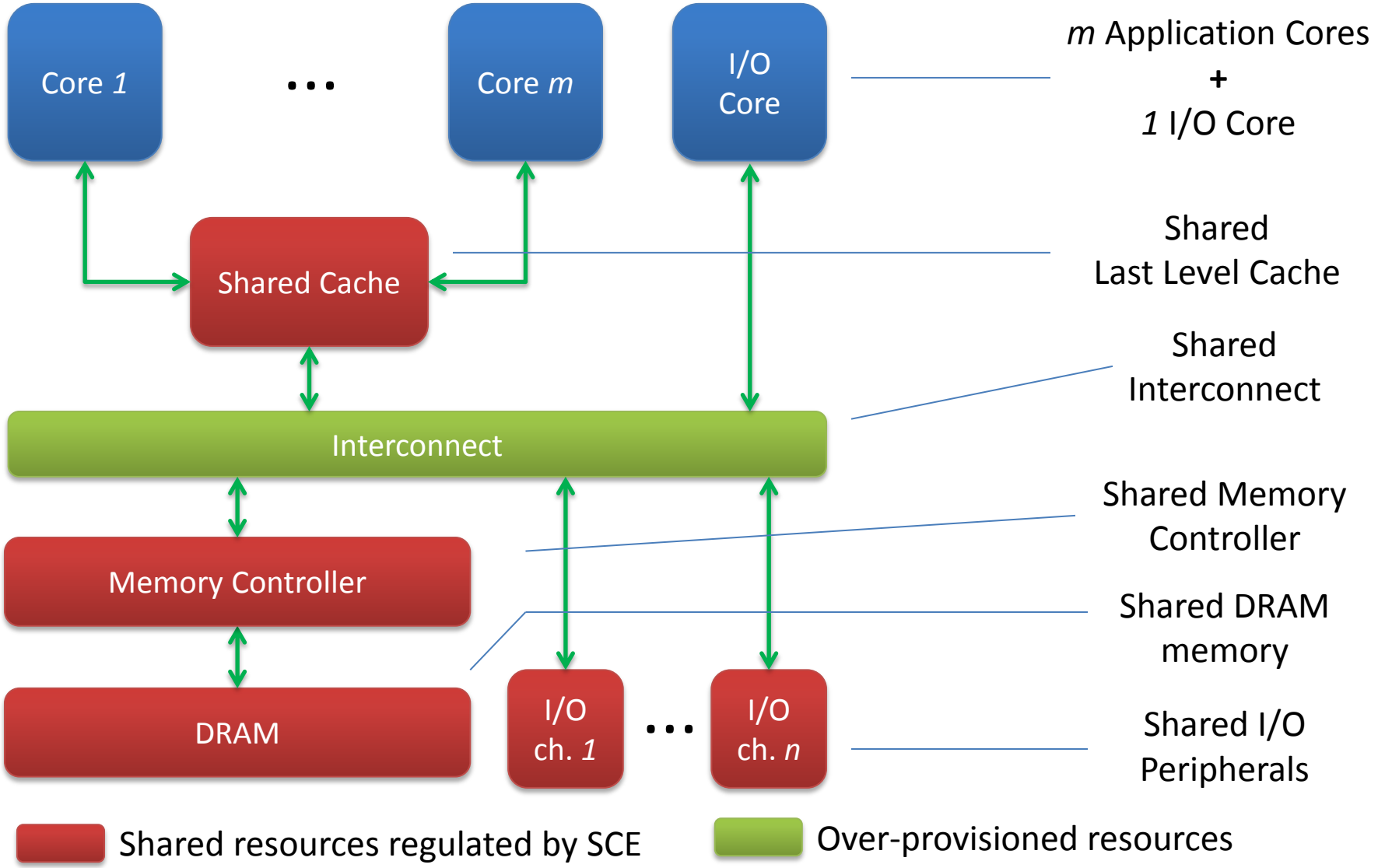
Sponsored in part by:

NSERC CRSNG

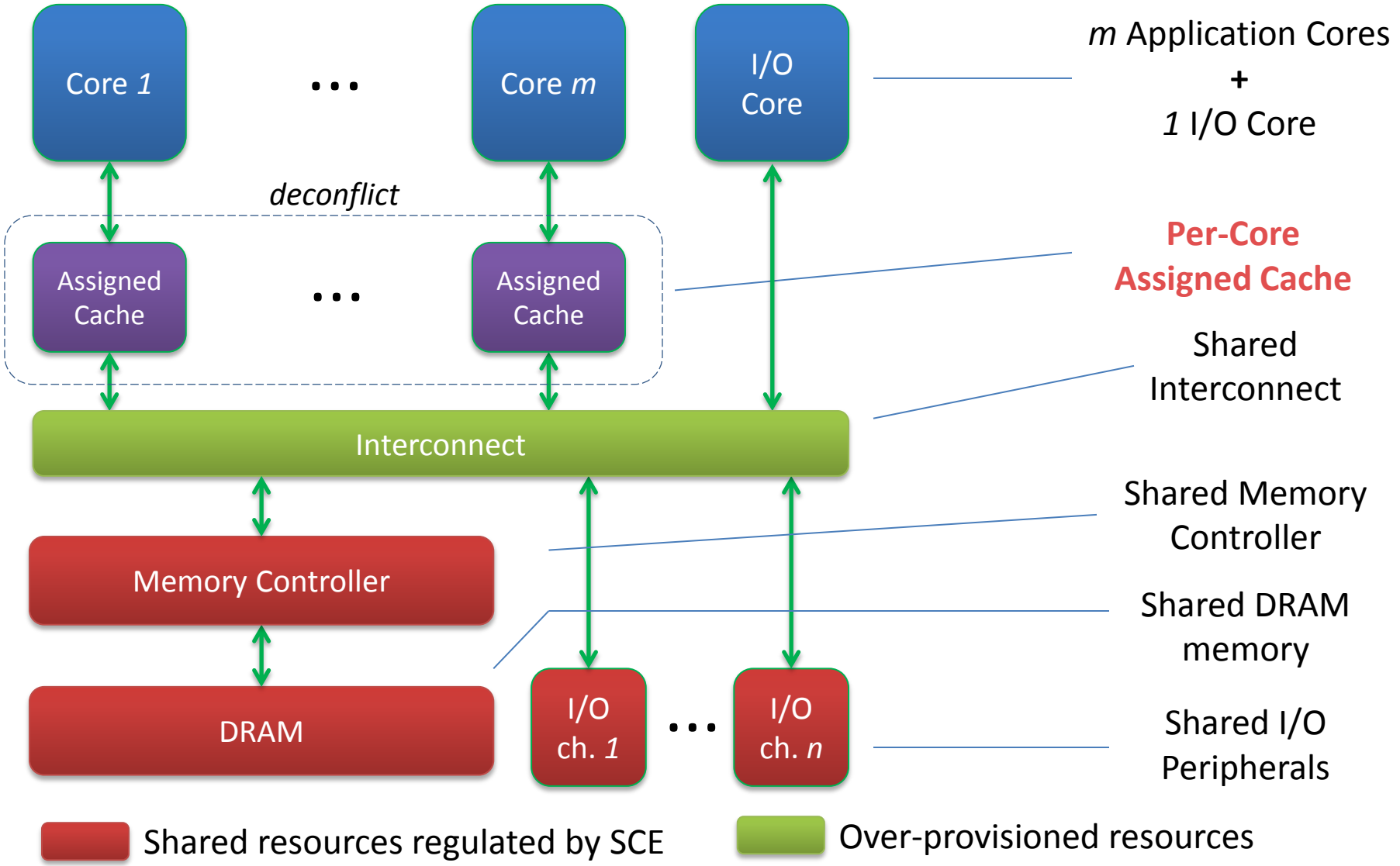
LOCKHEED MARTIN

Rockwell Collins

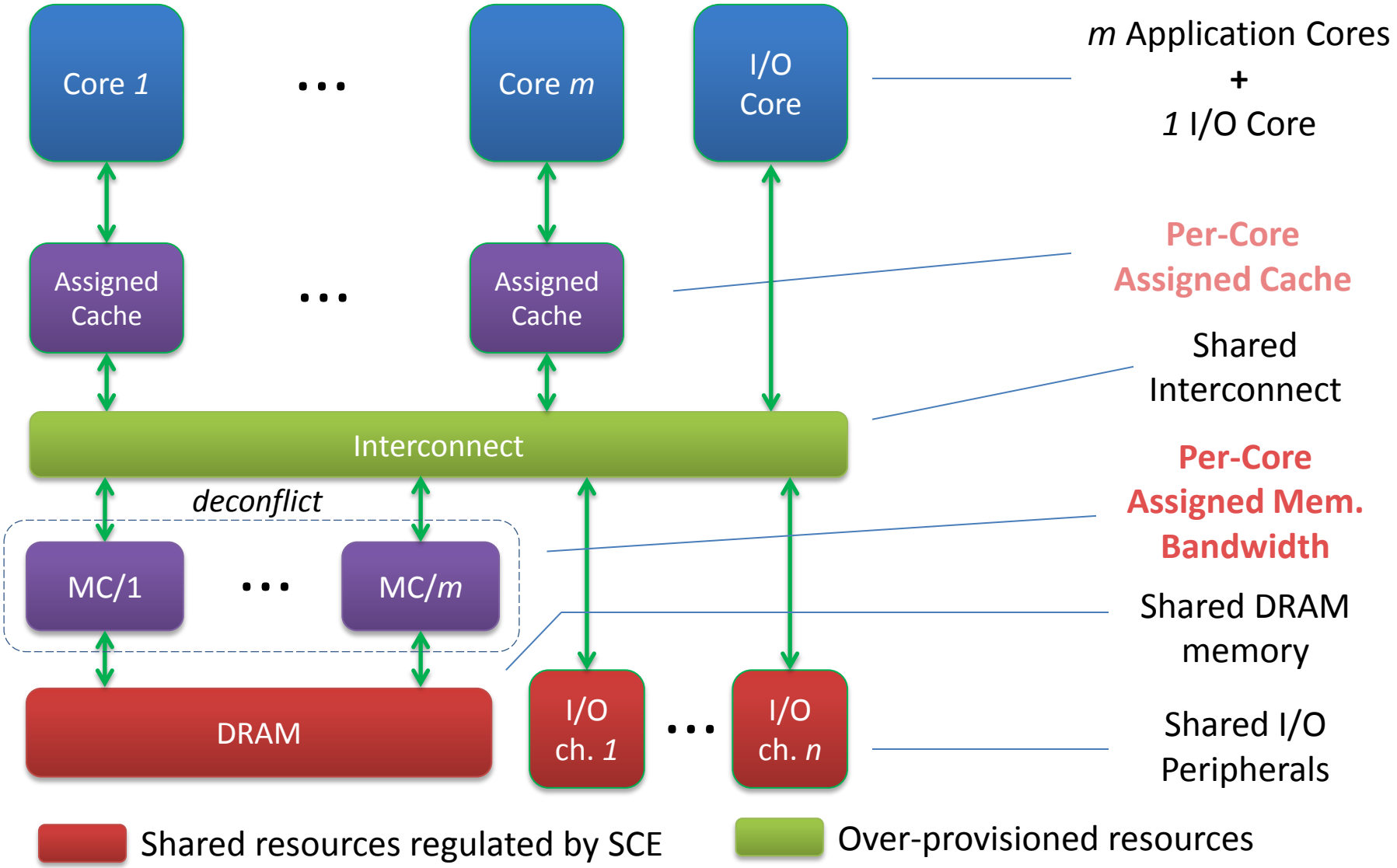
Shared Resources Regulated by SCE



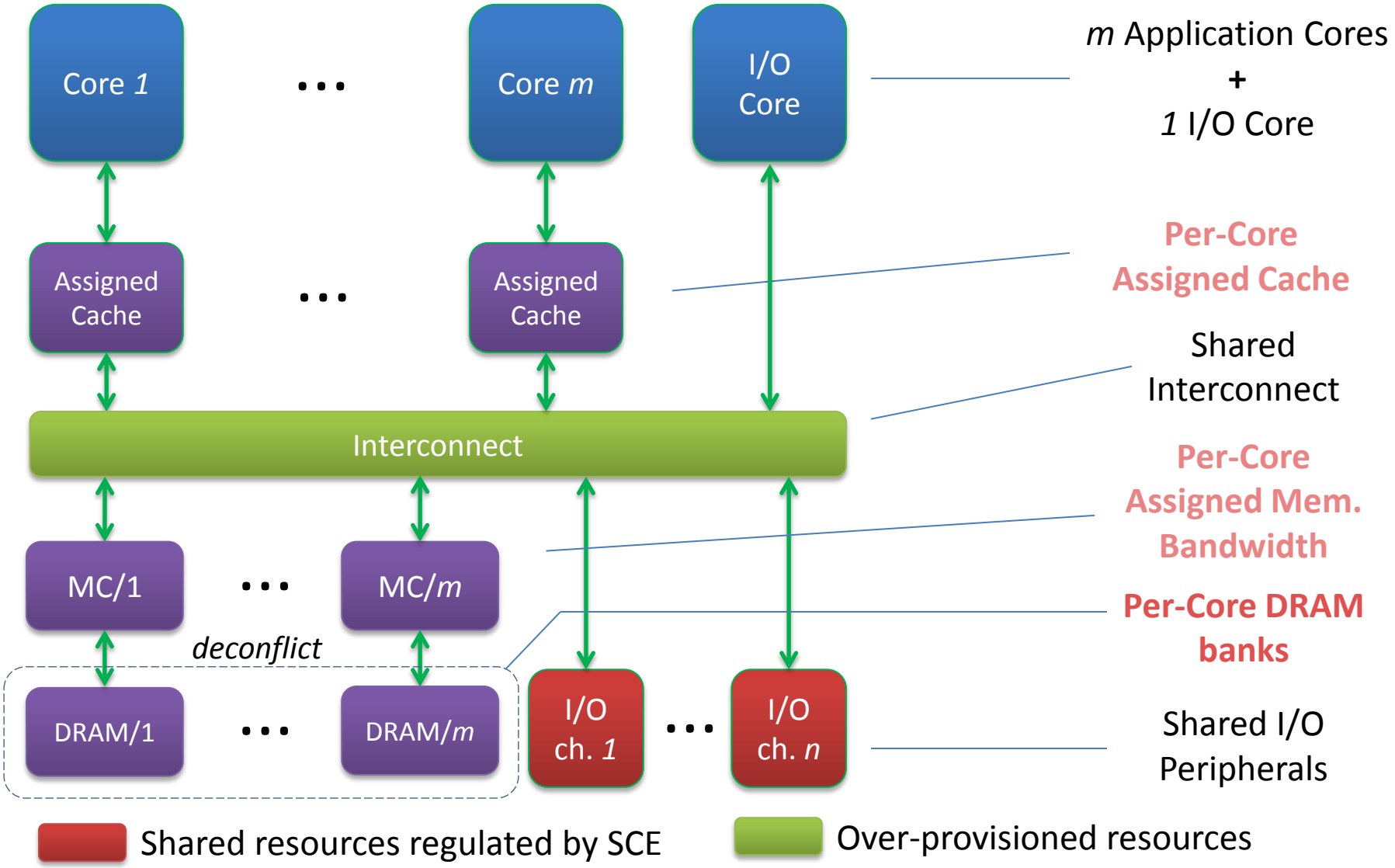
SCE Tech. 1 – Colored Lockdown



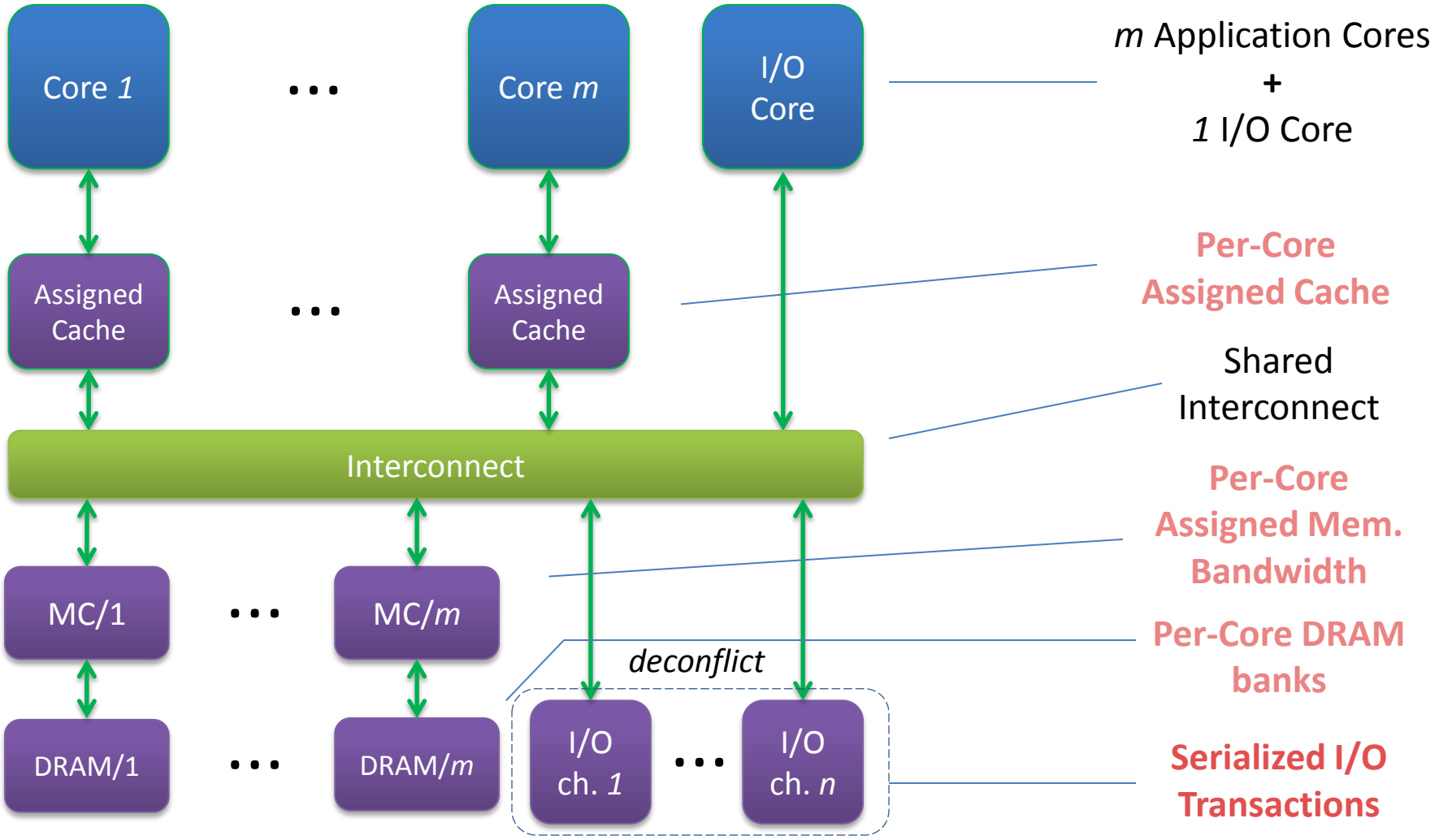
SCE Tech. 2 – MemGuard



SCE Tech. 2 – Palloc



SCE Tech. 4 – I/O Scheduling



SCE: Engineering Perspective

SCE dedicates $\frac{1}{m}$ of shared resources to each core.

WCET of tasks **directly depends** on the number of active cores *m*.

To certify for **up to *m*** active cores, find **WCET(*m*) for each task**

WCET(*m*) can be derived from **WCET calculated in isolation**

$$\mathrm{WCET}(m) = \mathrm{WCET}(1) + \hat{\mu} \cdot L_{size} \left(\frac{m}{BW_{min}} - \frac{1}{BW_{max}} \right) \quad *$$

(*) Renato Mancuso, Rodolfo Pellizzoni, Marco Caccamo, Lui Sha, Heechul Yun, WCET(m) Estimation in Multi-Core Systems using Single Core Equivalence. *In Proceedings of the 27th Euromicro Conference on Real-Time Systems* (ECRTS 2015), Lund, Sweden. To appear2015

SCE: Engineering Perspective

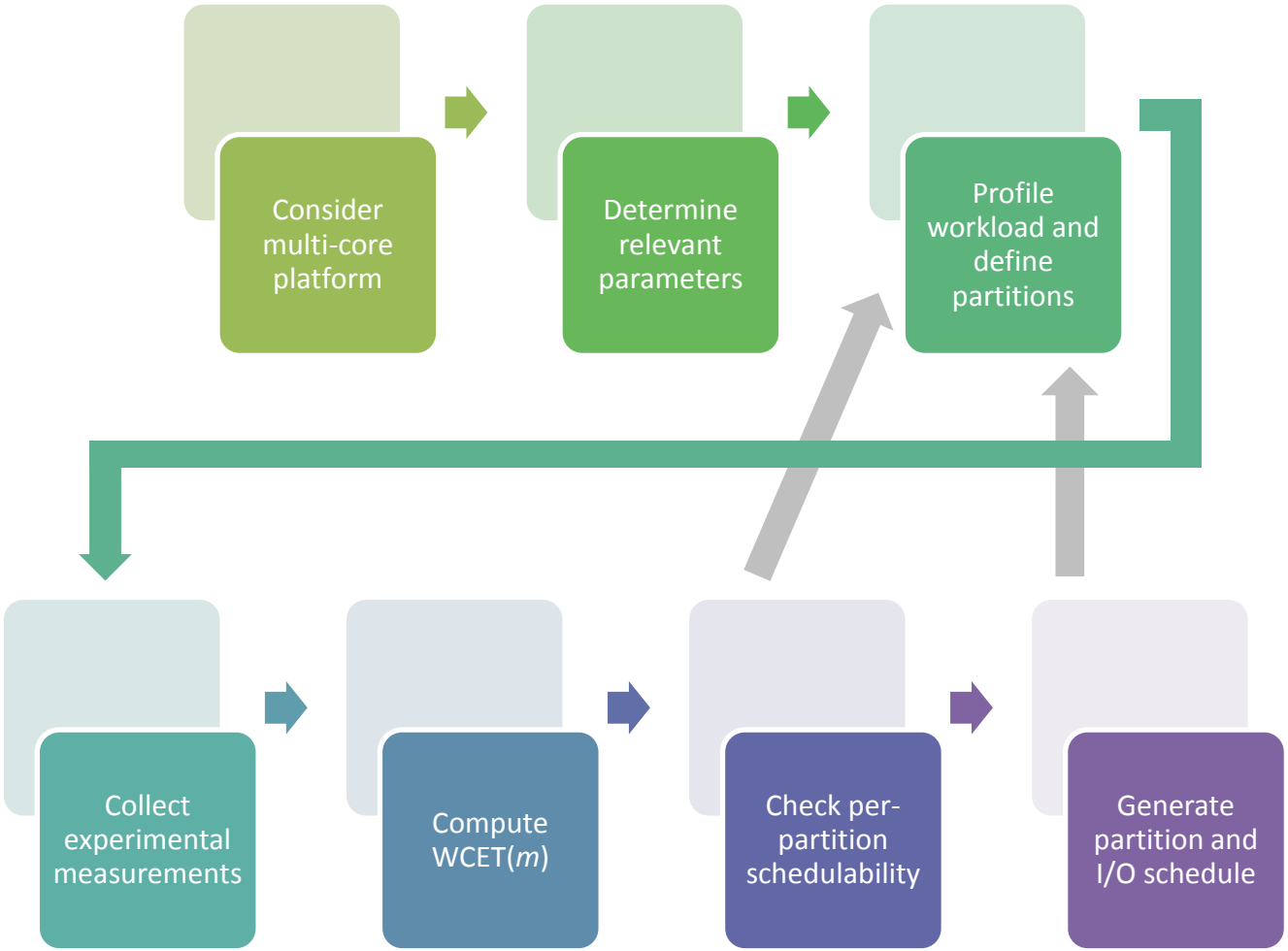
CMAS

the

SCE

single-core equivalence

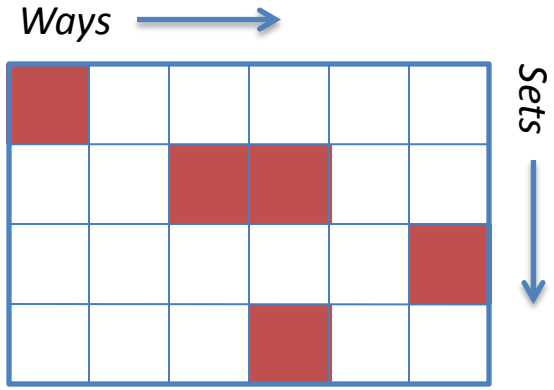
workflow



SCE: Colored Lockdown

CMAS

Our LLC Management Model: *



- Consider the LLC as a 2D array of lines
- Assign arbitrary sets of blocks to tasks

✔ Addresses all the sources of interference

✔ Converts the LLC cache in a deterministic object

at the granularity of a single memory page

✔ Allows the use of legacy code

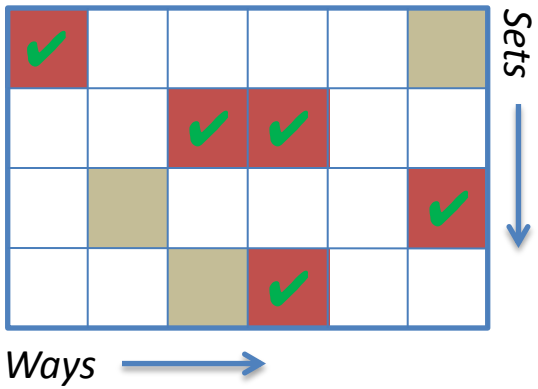
✔ Provides flexibility in cache assignment

(*) Renato Mancuso, Roman Dudko, Emiliano Betti, Marco Cesati, Marco Caccamo, Rodolfo Pellizzoni, Real-Time Cache Management Framework for Multi-Core Architectures. *In Proceedings of the 19th IEEE International Conference on Real-Time and Embedded Technology and Applications Symposium* (RTAS 2013), Philadelphia, PA, USA.

Colored Lockdown: Profiling

CMAS

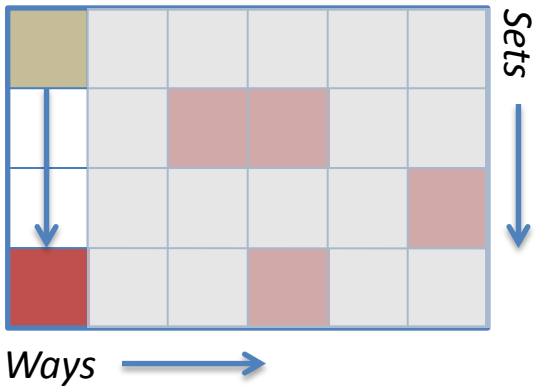
1. Profiling



- Aims at using the cache **deterministically**
- Has to deal with **limited cache size**
- Run task in **sandbox,** analyze **memory accesses**
- Find **frequently accessed** (**hot**) memory regions

Colored Lockdown: Coloring

2. Coloring

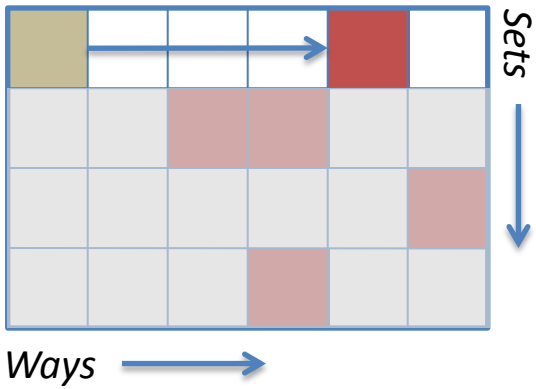


- Leverages on the **virtual** → **physical** translation layer
- Used to **move page mapping** across sets (up/down)
- Transparent to the **programmer**
- Transparent to the **application**

Colored Lockdown: Lockdown

CMAS

3. Lockdown



- Uses architecture-specific **lockdown features**
- Used to **allocate pages** on selected ways (left/right)
- Can be implemented at **OS-level**

Summary

- DO 178C can be used for multicore, only if each core in a multicore chip is logically equivalent to a single core chip (SCE). **SCE enables to modularly certify** software **one core at a time** using DO 178C.
- Technologies to implement the SCE framework are open to innovation. However**, violation of SCE objective** means that we would allow the modification of applications in one core to **"decertify"** the applications in other cores.
- Challenges in SCE technology development and certification
  - Currently SCE addresses the **isolation challenges**.
    - **Intercore communication** support needs to be completed.
    - How to use more than one core for **big applications** needs to be completed.
  - SCE certification is **architecture dependent**. Validated hardware abstraction required.
  - SCE integrates with low level RTOS operation and is harder to verify than application level software. **But only needs to be done once for a platform.**

Single Core Equivalence Framework (SCE) For Certifiable Multicore Avionics

Memory Mnagement

a collaboration of:

Presenter: Heechul Yun

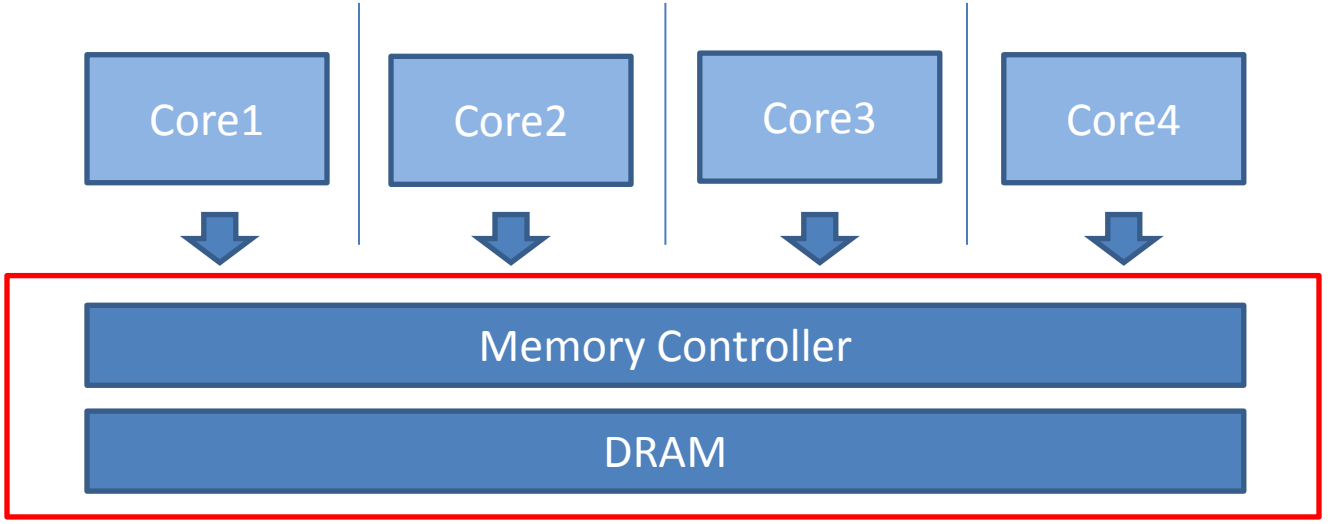
Sponsored in part by:

NSERC CRSNG

LOCKHEED MARTIN

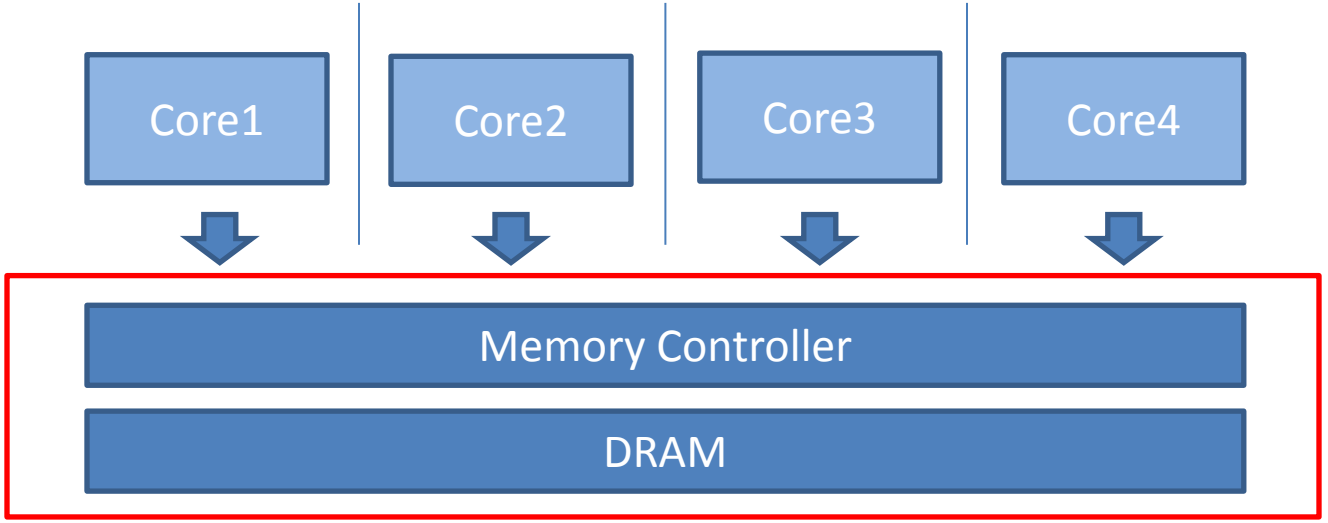
Rockwell Collins

This Talk



- Focus on **DRAM** and **memory controller**
- Present **SW mechanisms** for timing predictability

Why Important?

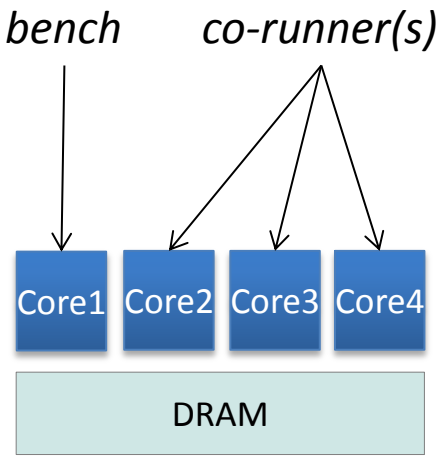


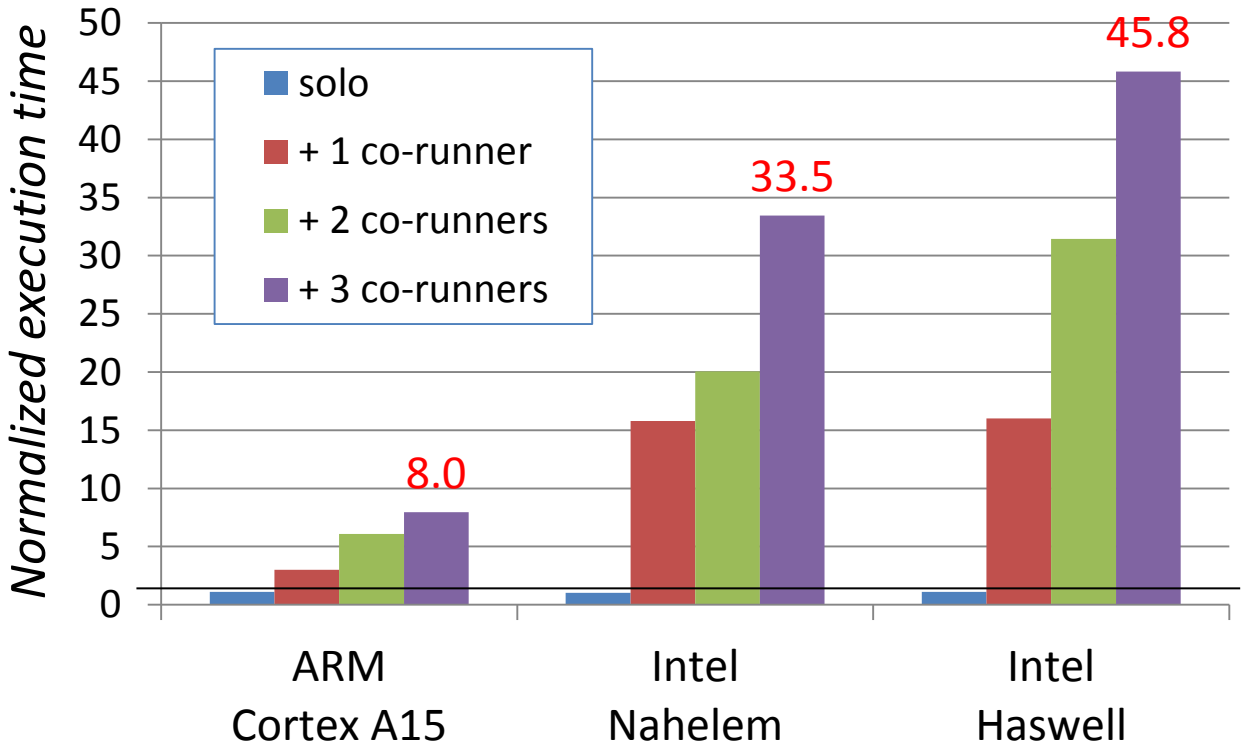
- Memory is becoming a bottleneck
- Performance is very **poor** in the **worst-case**

How Serious?

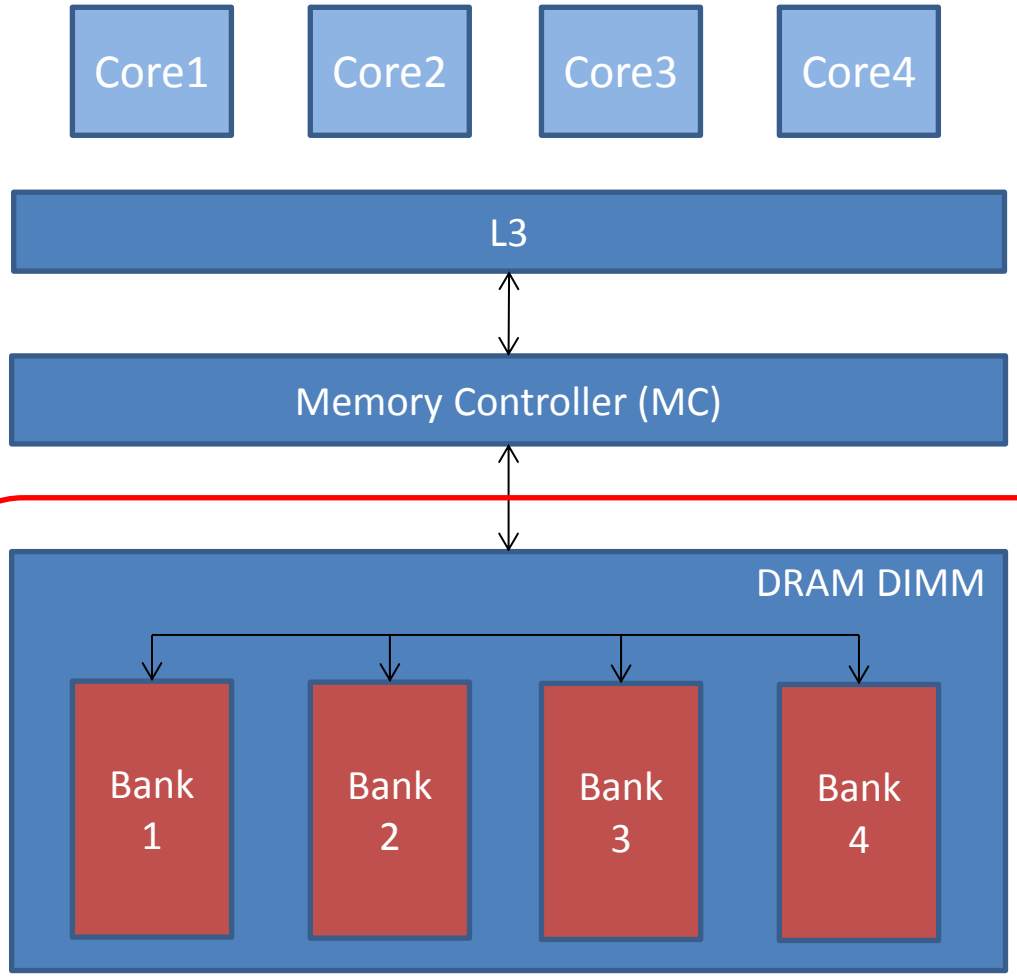
- Synthetic worst-case experiments:

Up to 45.8X slowdown



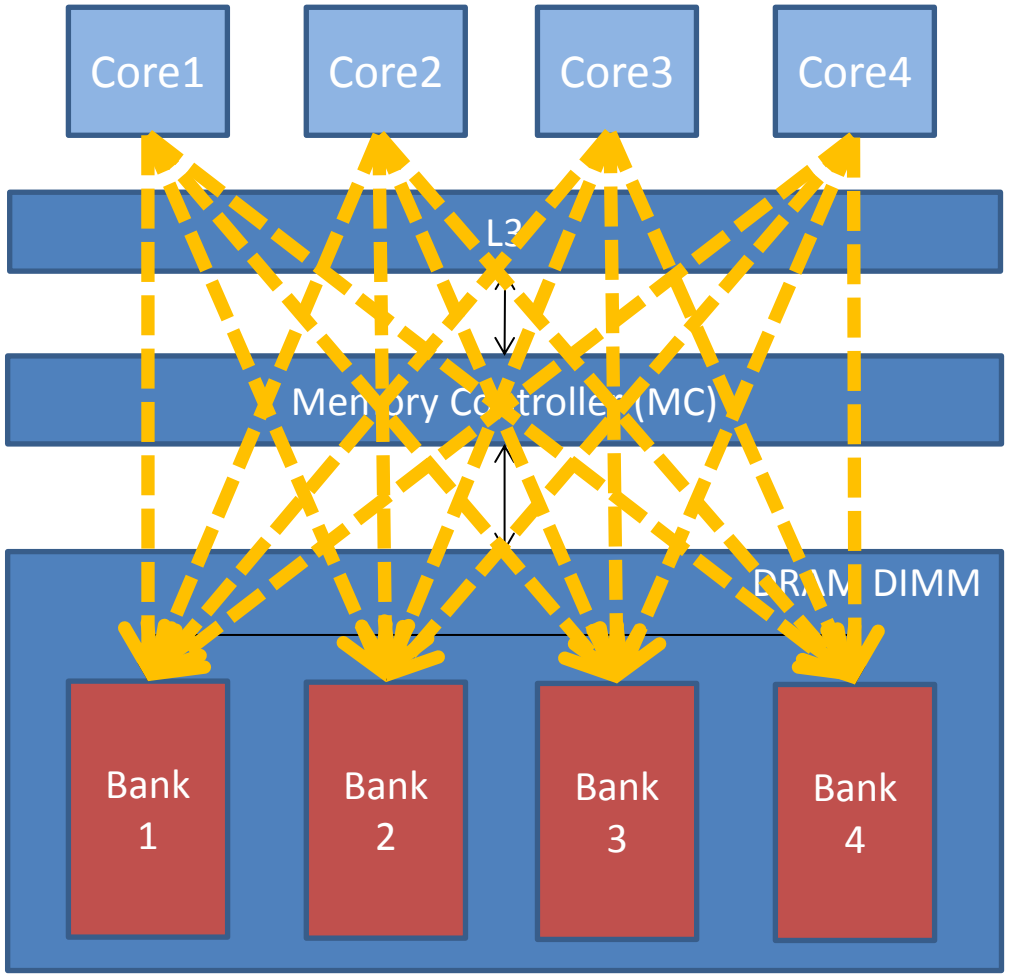


Background: DRAM Organization



- Have multiple banks
  - 8 ~ 16 banks per DIMM
- Different banks can be accessed in parallel

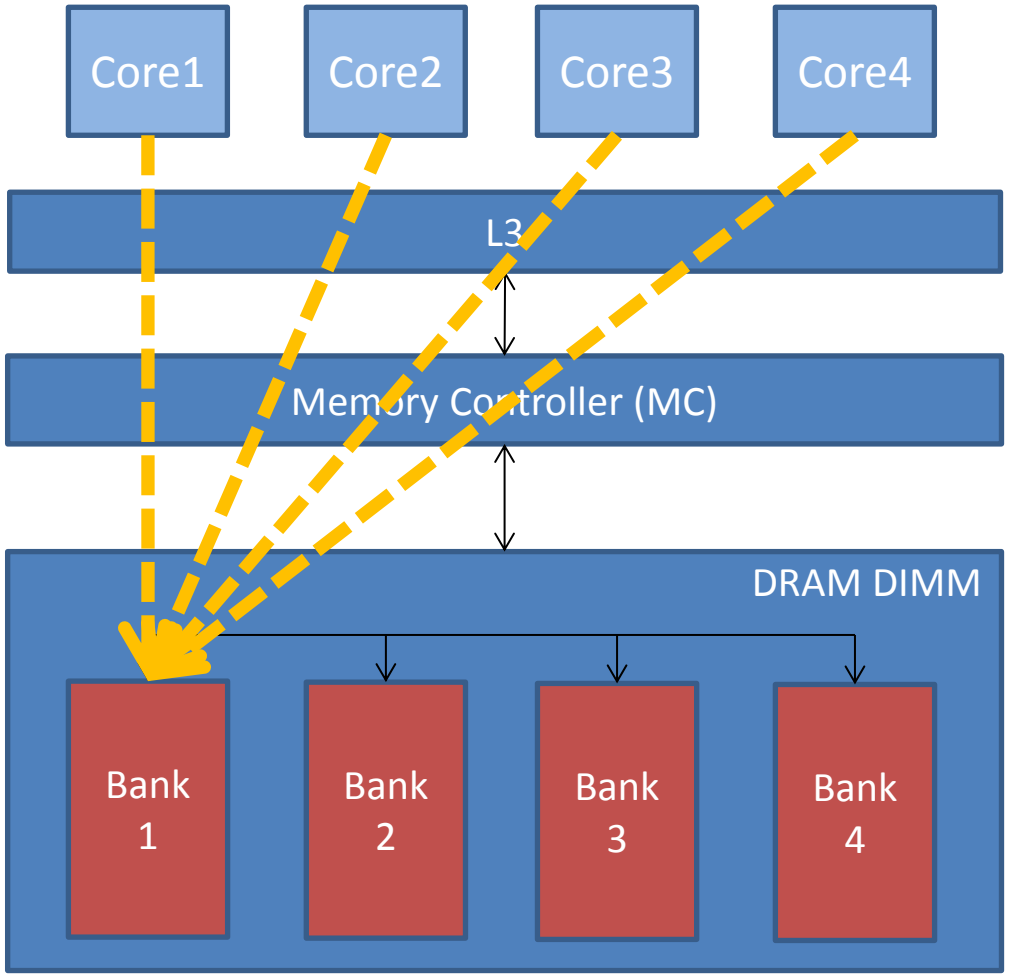
Most-cases



Mess

- Performance = ??

Worst-case



Slow

- 1bank b/w
  - Less than peak b/w
  - How much?

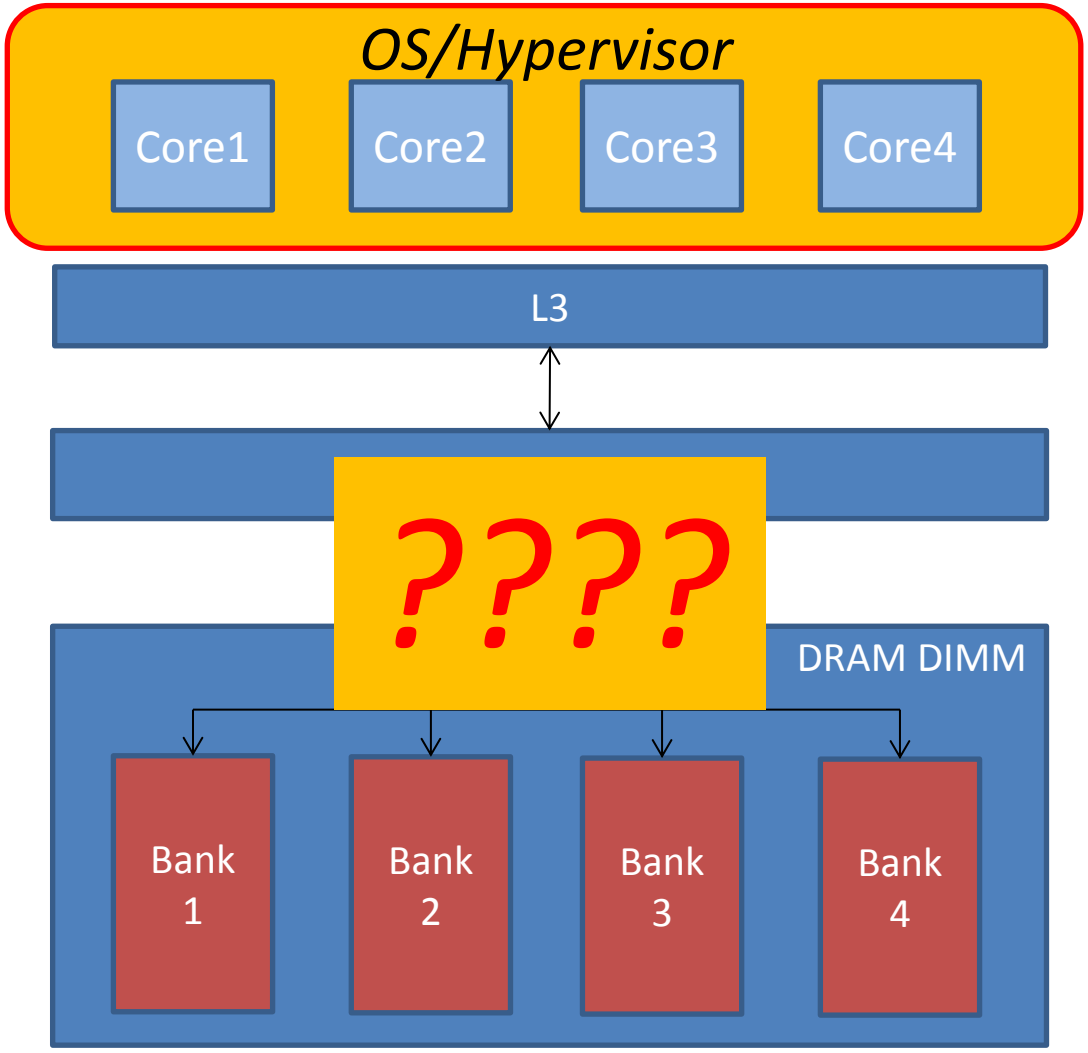
Outline

- Introduction
- DRAM Background
- **Control mechanisms**
  - **PALLOC: Space (bank) partitioning ***
  - **MemGuard: Bandwidth partitioning ****
- Conclusion

(*) Heechul Yun, Renato Mancuso, Zheng-Pei Wu, Rodolfo Pellizzoni. PALLOC: DRAM Bank-Aware Memory Allocator for Performance Isolation on Multicore Platforms. *IEEE Intl. Conference on Real-Time and Embedded Technology and Applications Symposium (RTAS)*, IEEE, 2014

(**) Heechul Yun, Gang Yao, Rodolfo Pellizzoni, Marco Caccamo, and Lui Sha. MemGuard: Memory Bandwidth Reservation System for Efficient Performance Isolation in Multi-core Platforms. *IEEE Intl. Conference on Real-Time and Embedded Technology and Applications Symposium (RTAS)*, IEEE, 2013.

Problem

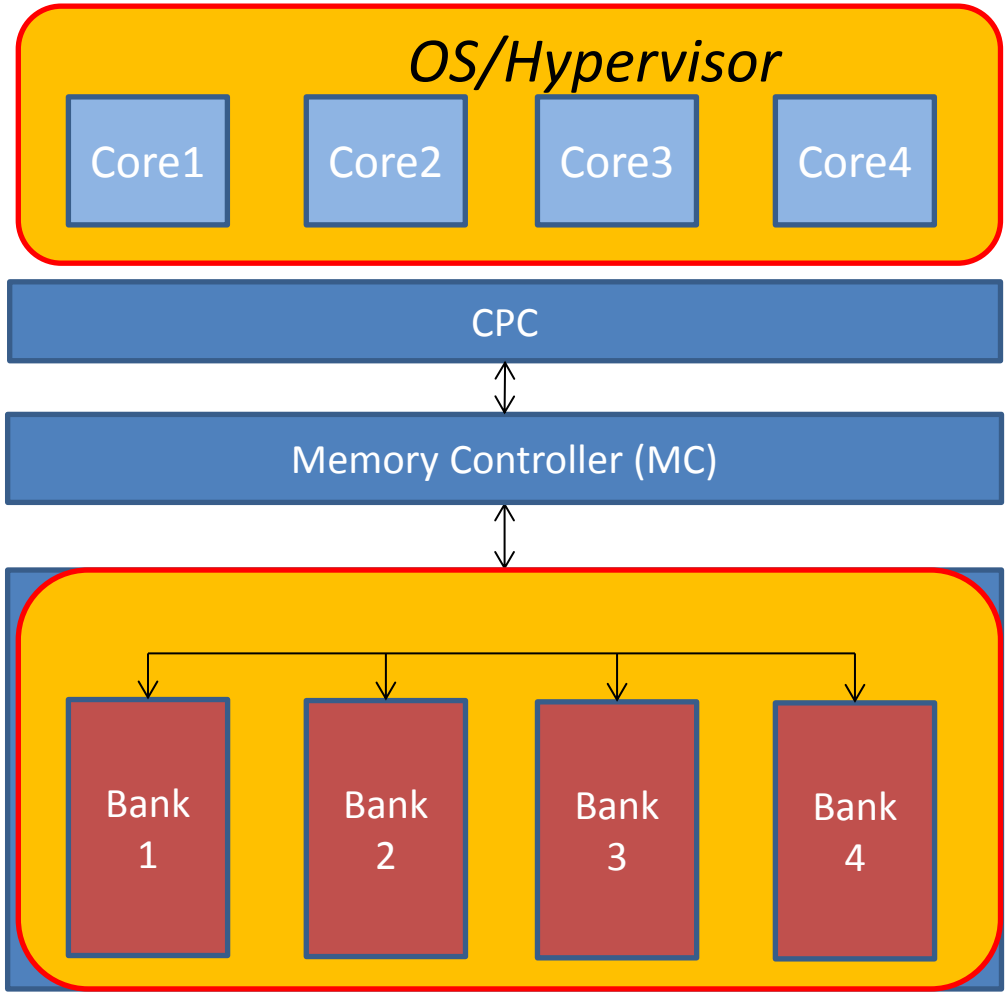


- OS/hypervisor is **unaware of DRAM banks**
- Memory pages are spread all over multiple banks

Unpredictable

Bank Conflict

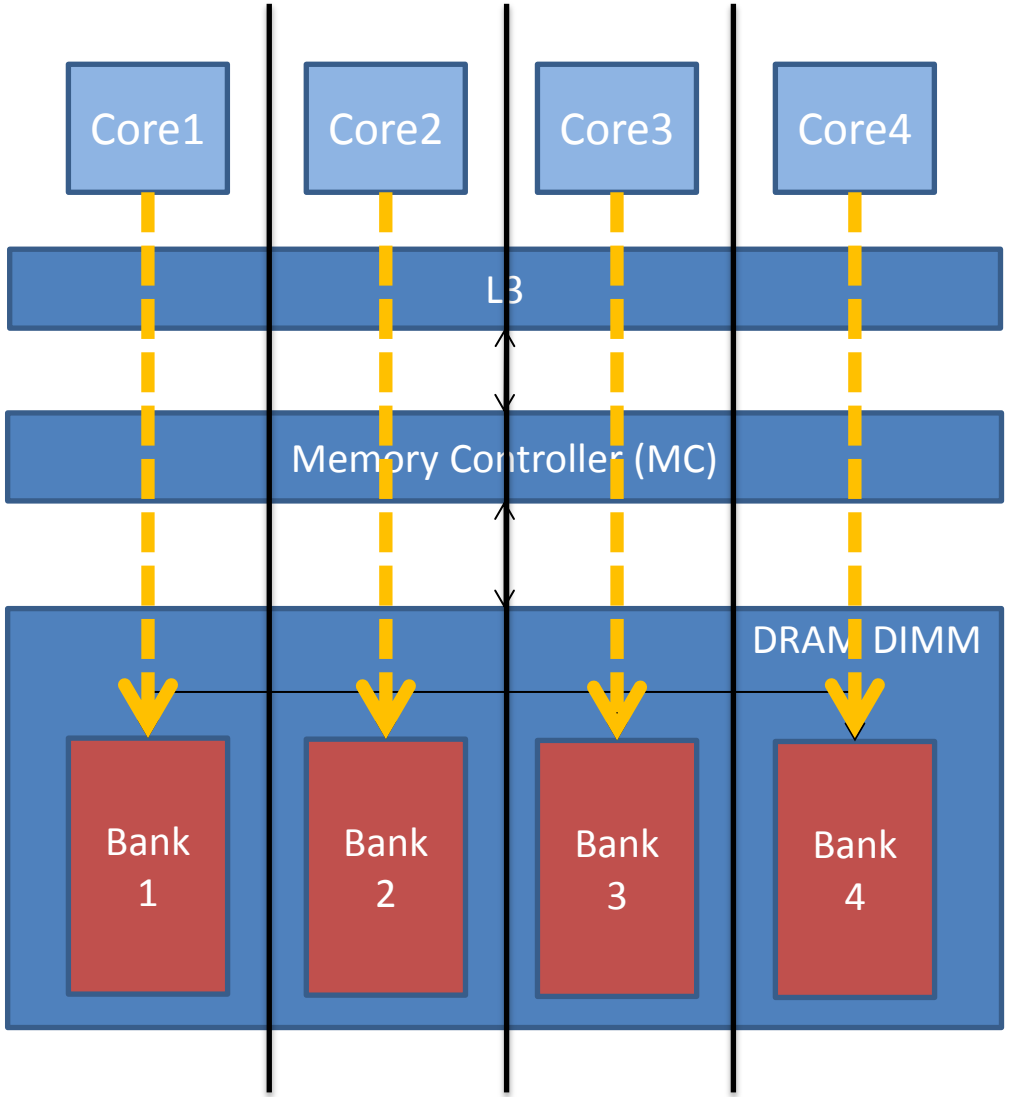
PALLOC



- Aware of DRAM mapping
- Each page can be allocated to a desired DRAM bank

Flexible Allocation Policy

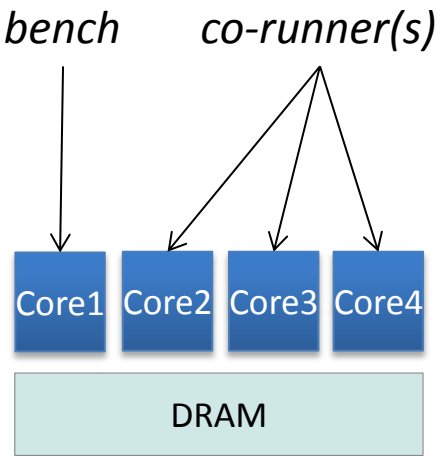
PALLOC

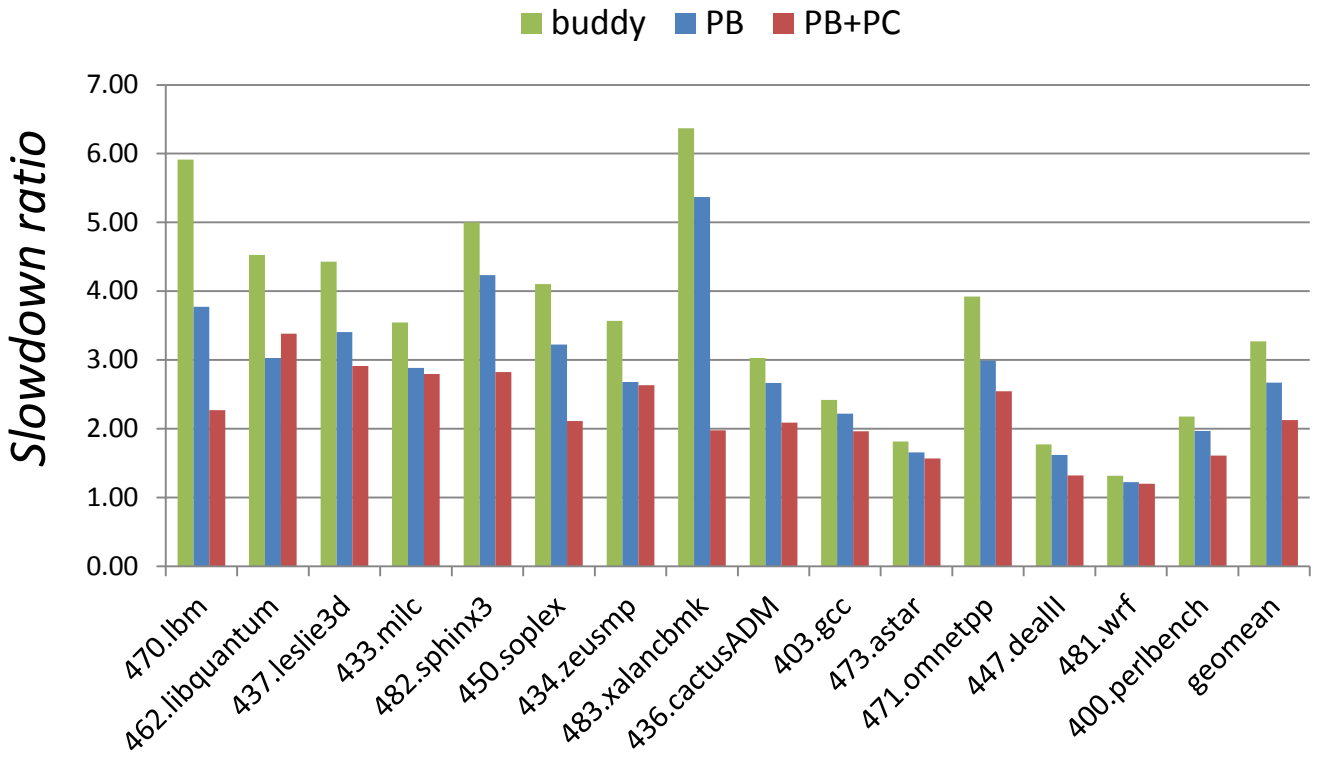


- Private banking
  - *Allocate* pages on certain *exclusively* assigned banks

Better Performance Isolation

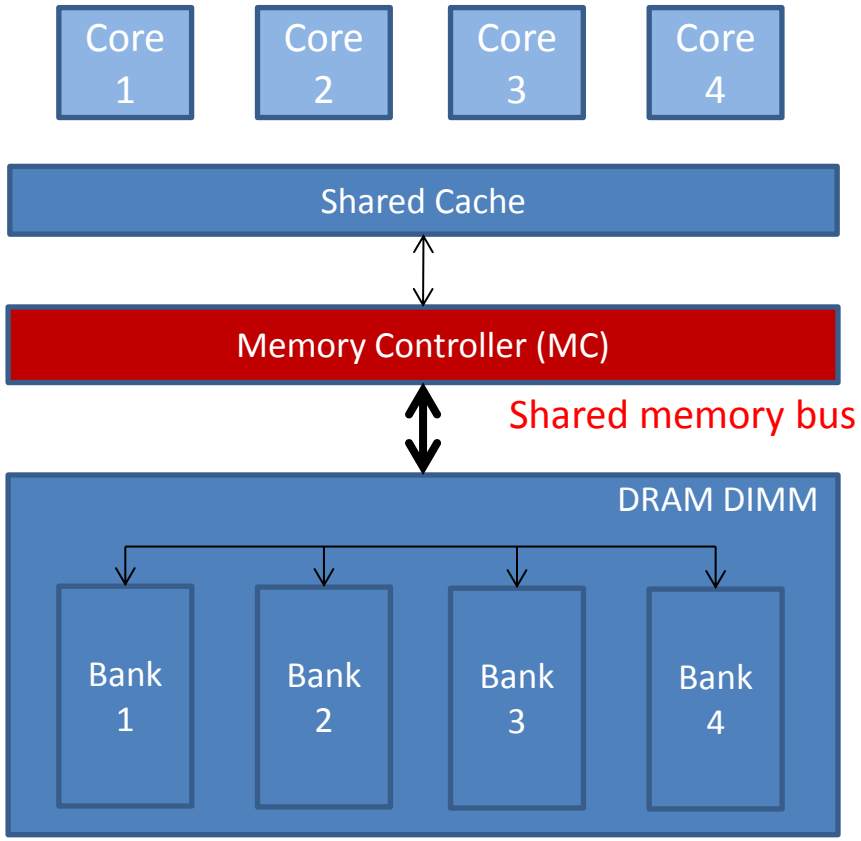
Performance Slowdown





- PB: DRAM bank partitioning only;
- PB+PC: DRAM bank and Cache partitioning
- Bank (and cache) partitioning improves isolation, **but far from ideal**
  - Due to **Memory bus bandwidth contention (next technique)**

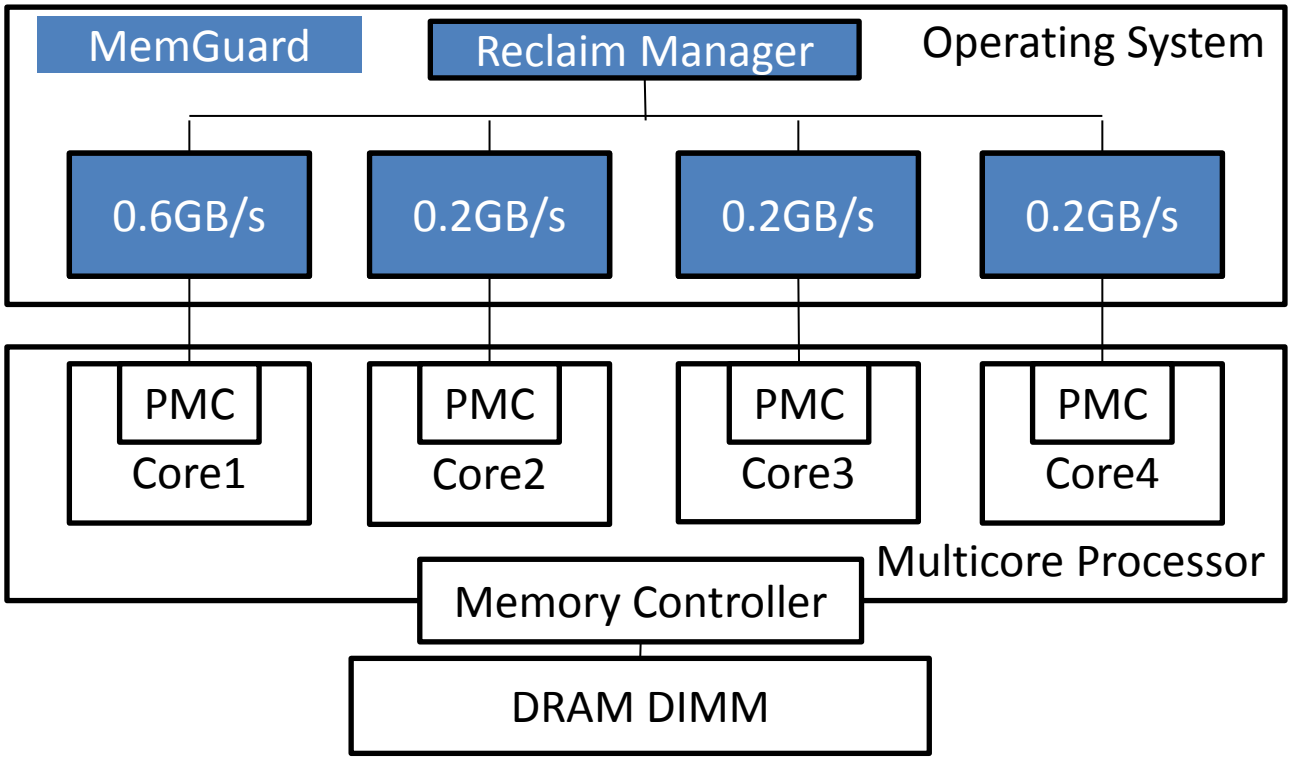
Problem



- Banks can be accessed in parallel
- But all banks **share a memory bus**
- Memory bandwidth **<<** CPU demands

Memory bandwidth contention

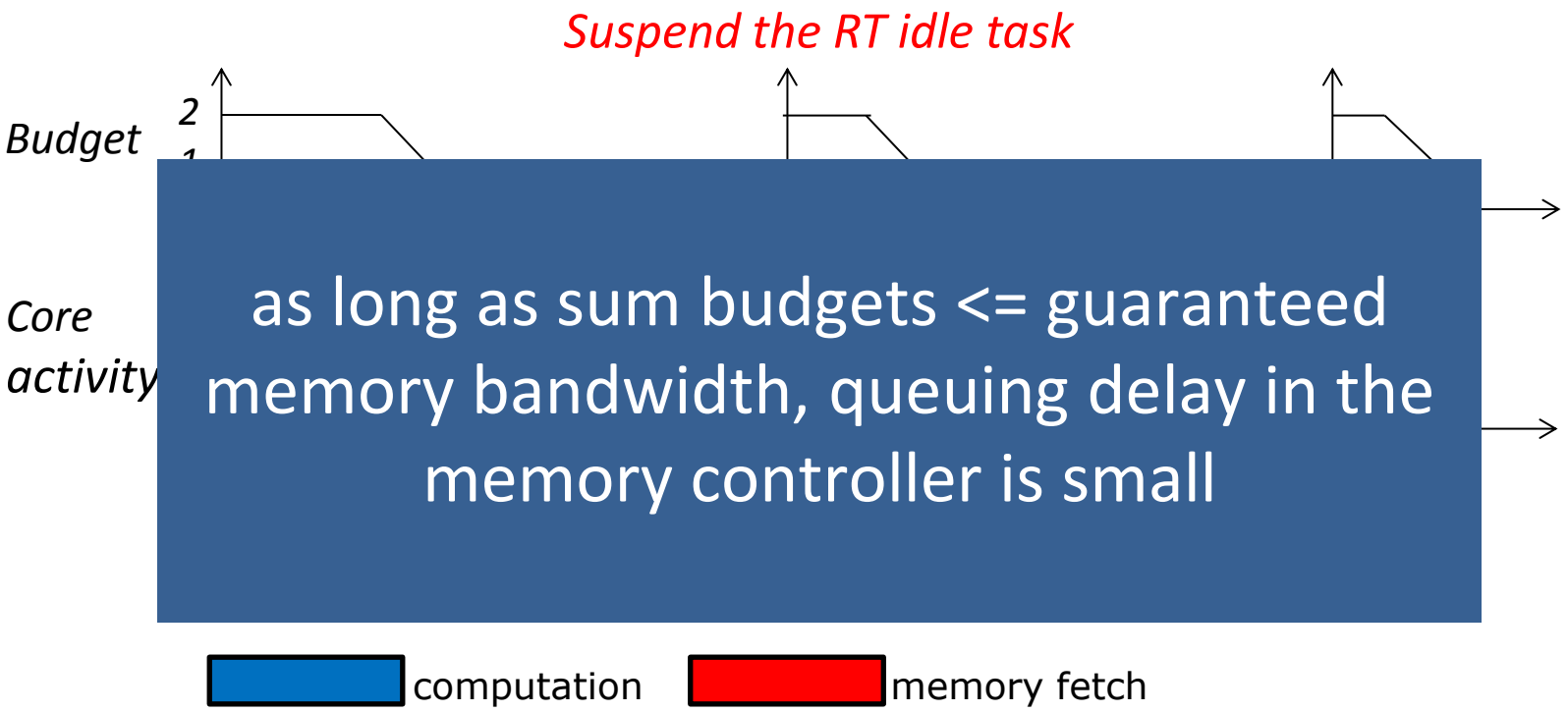
MemGuard



- Goal: guarantee *minimum memory b/w* for each core
- How: b/w reservation

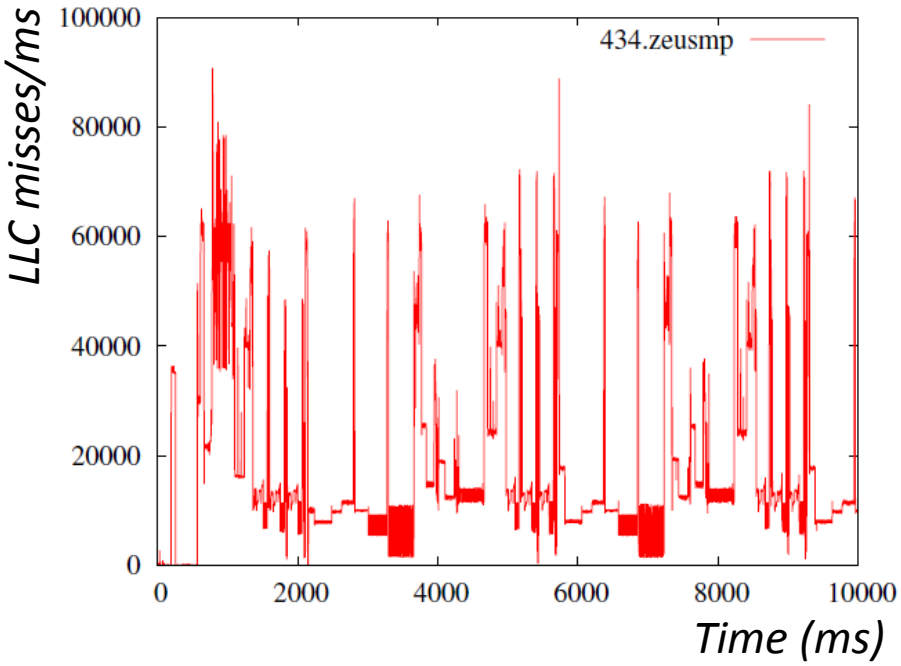
Reservation

- Idea
  - Reserve per-core memory bandwidth via the OS scheduler
    - Use h/w PMC to monitor memory request rate

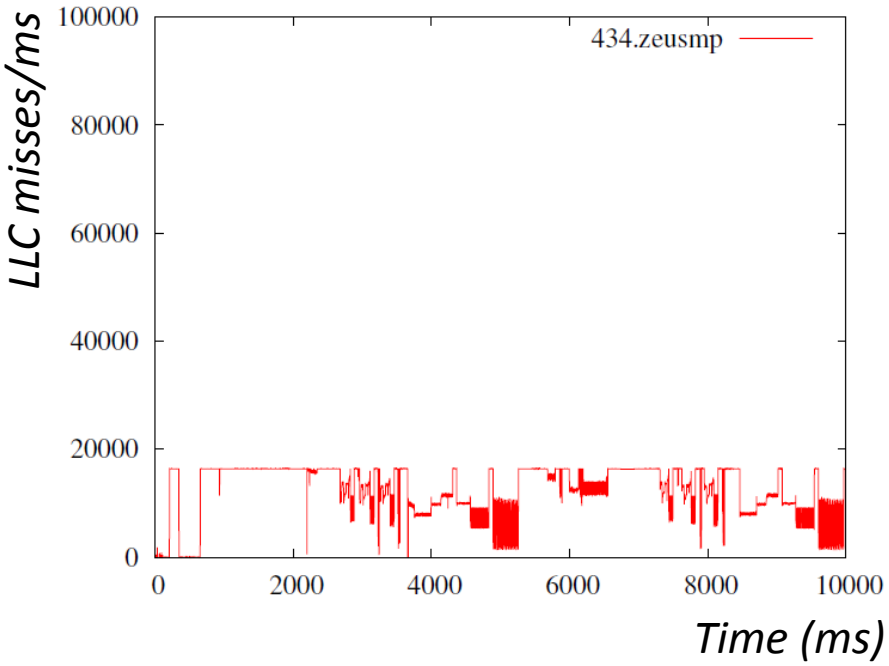


Impact of Reservation

W/o MemGuard



MemGuard (1GB/s)



Conclusion

- Multicore certification is a huge challenge
- Main memory is an important interference channel
  - Bank (space) conflict
  - Bandwidth contention
- Proposed control mechanisms
  - PALLOC: DRAM bank (space) control
  - MemGuard: DRAM bandwidth (time) control

  →*Improved performance isolation*

Single Core Equivalence Framework (SCE) For Certifiable Multicore Avionics

IMA & I/O Management

a collaboration of:

Presenter: Jung-Eun Kim

Sponsored in part by:

NSERC CRSNG

LOCKHEED MARTIN

Rockwell Collins

The MCP IMA Challenge

- *"Authorities are not currently aware of any MCP hardware and software implementations … in the way … currently ensured for the applications of an IMA on a single core processor (SCP)."* *in CAST 32*
- *"This paper may be extended in future to address MCPs with more than two active cores and MCP IMA implementations."* *in CAST 32*

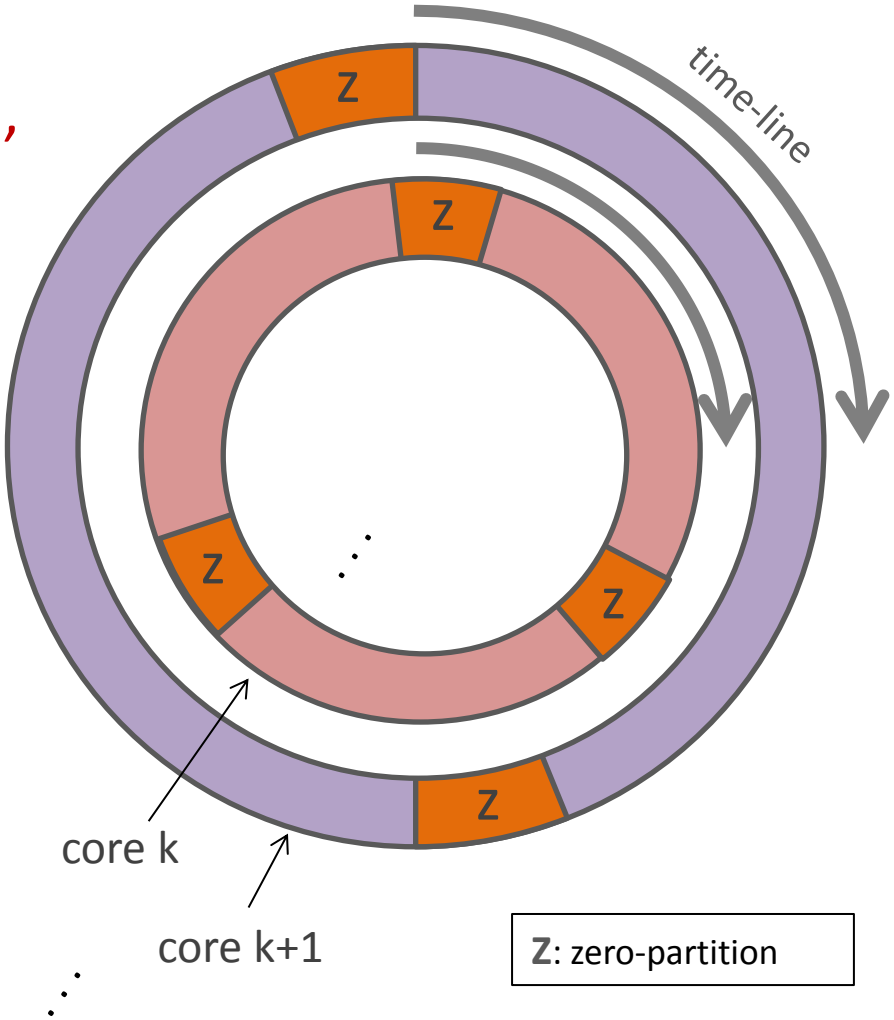
Migration: I/O Conflicts

– Zero-partition

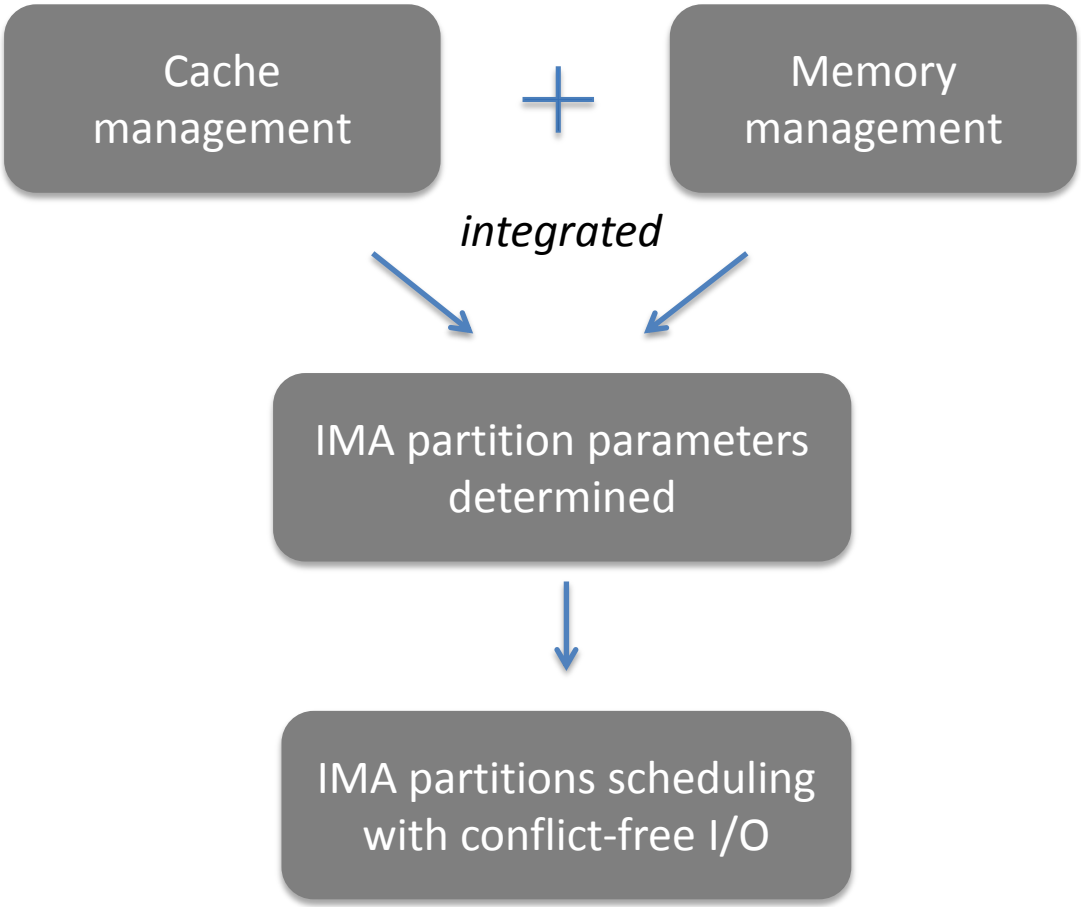
: a special-purpose 'I/O partition'

– Migrating multiple single-core IMAs to a multicore system.

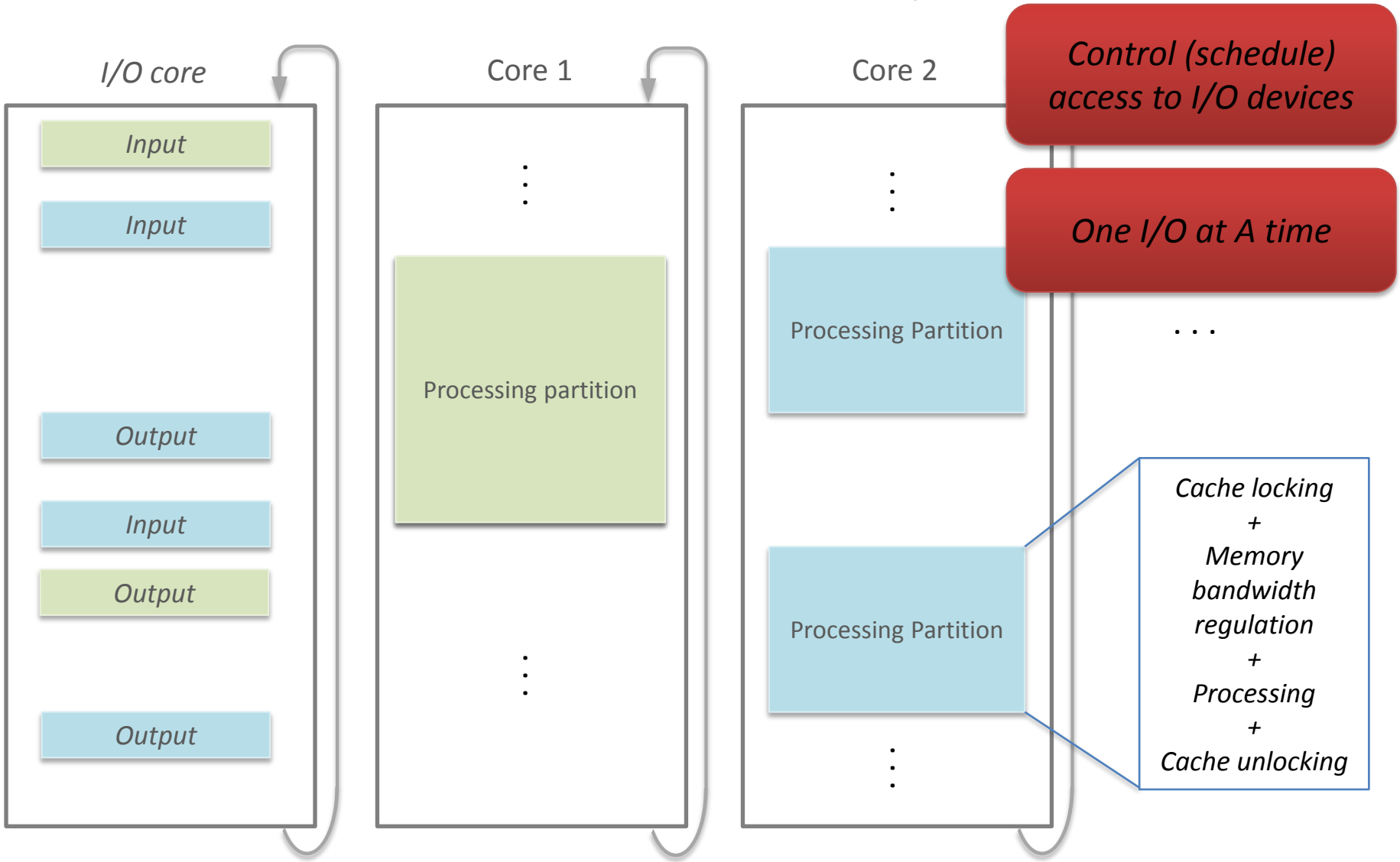
- Multiple rate groups
- Shared I/O channel conflicts
- Synchronizing challenge



All Things are Putting Together

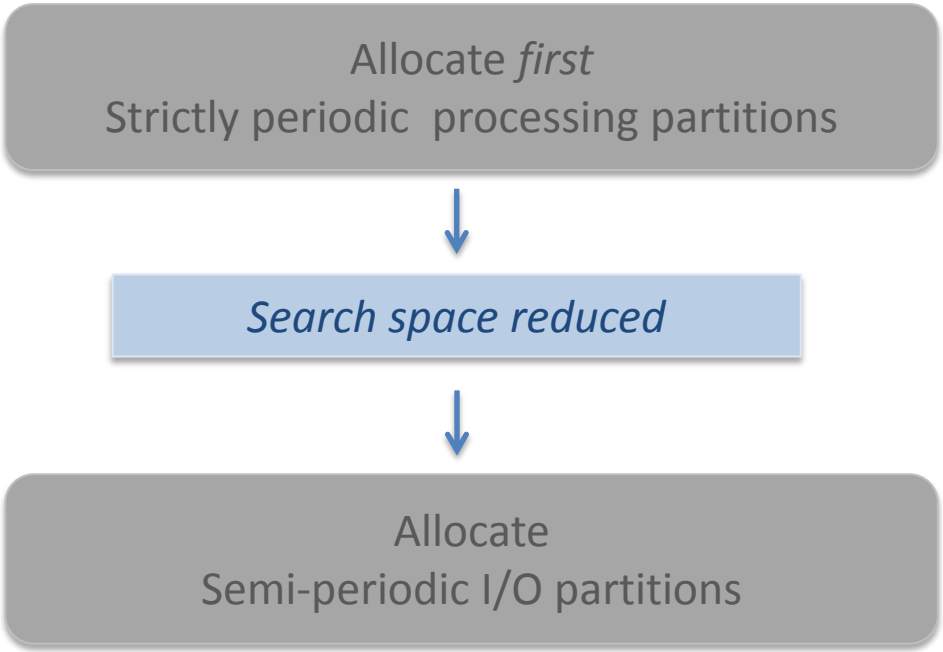


Generating IMA Partition Scheduling for Conflict-free I/O



Idea – How to Solve

Bottleneck-first approach

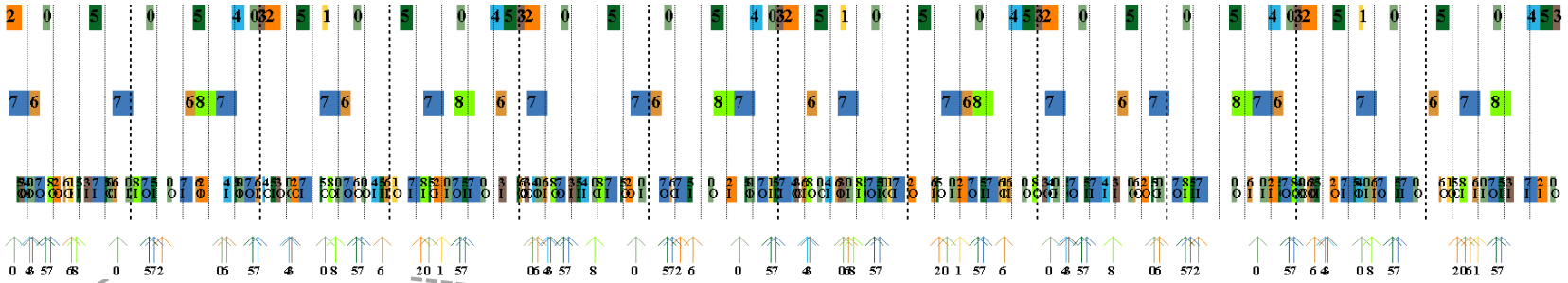


Jung-Eun Kim, Man-Ki Yoon, Richard Bradford and Lui Sha, "*Integrated Modular Avionics (IMA) Partition Scheduling with Conflict-Free I/O for Multicore Avionics Systems*," in Proceedings of the 38th IEEE Computer Software and Applications Conference (COMPSAC 2014), Jul. 2014.

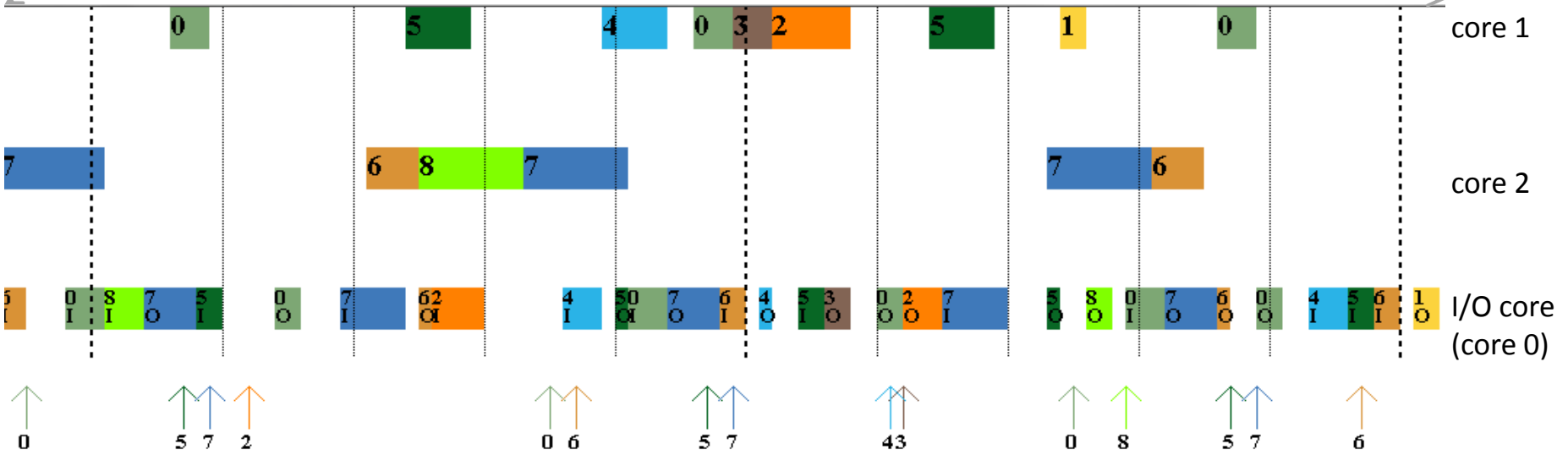
Jung-Eun Kim, Man-Ki Yoon, Sungjin Im, Richard Bradford and Lui Sha, "*Optimized Scheduling of Multi-IMA Partitions with Exclusive Region for Synchronized Real-Time Multi-Core System*," in Proceedings of the 16th ACM/IEEE Design, Automation, and Test in Europe (DATE 2013), pp. 970-975, Mar. 2013.

Result of a Practical Example

1 I/O Core + 2 Processing cores; Periods (core_1: 40,200,100,100,100,40; core_2: 60, 40, 100); LCM=600



(magnified)



D. Locke, L. Lucas and J. Goodenough, "Generic avionics software specification," Software Engineering Institute, Pittsburgh, Pennsylvania,1990, CMU/SEI-90-

SCE Summary

- **DO 178C can be used for multicore, only if each core in a multicore chip is logically equivalent to a single core chip (SCE).** That is, intercore interferences can be certifiably bounded and for all core workload configurations.
- **SCE Technologies** is **open to innovation**. However**, violation of SCE objective** means that we would allow the modification of applications in one core to **"decertify"** the applications in other cores.
- Challenges in SCE technology development and certification
  - Currently SCE addresses the isolation challenges.
    - **Intercore communication** support needs to be completed.
    - How to use more than one core for **big applications** needs to be completed.
  - SCE certification **is chip architecture dependent**, requires hardware primitives currently found in some Freescale chips.
  - **Validated hardware abstraction** required.
  - **Verification and certification** of SCE design & implementation are required.

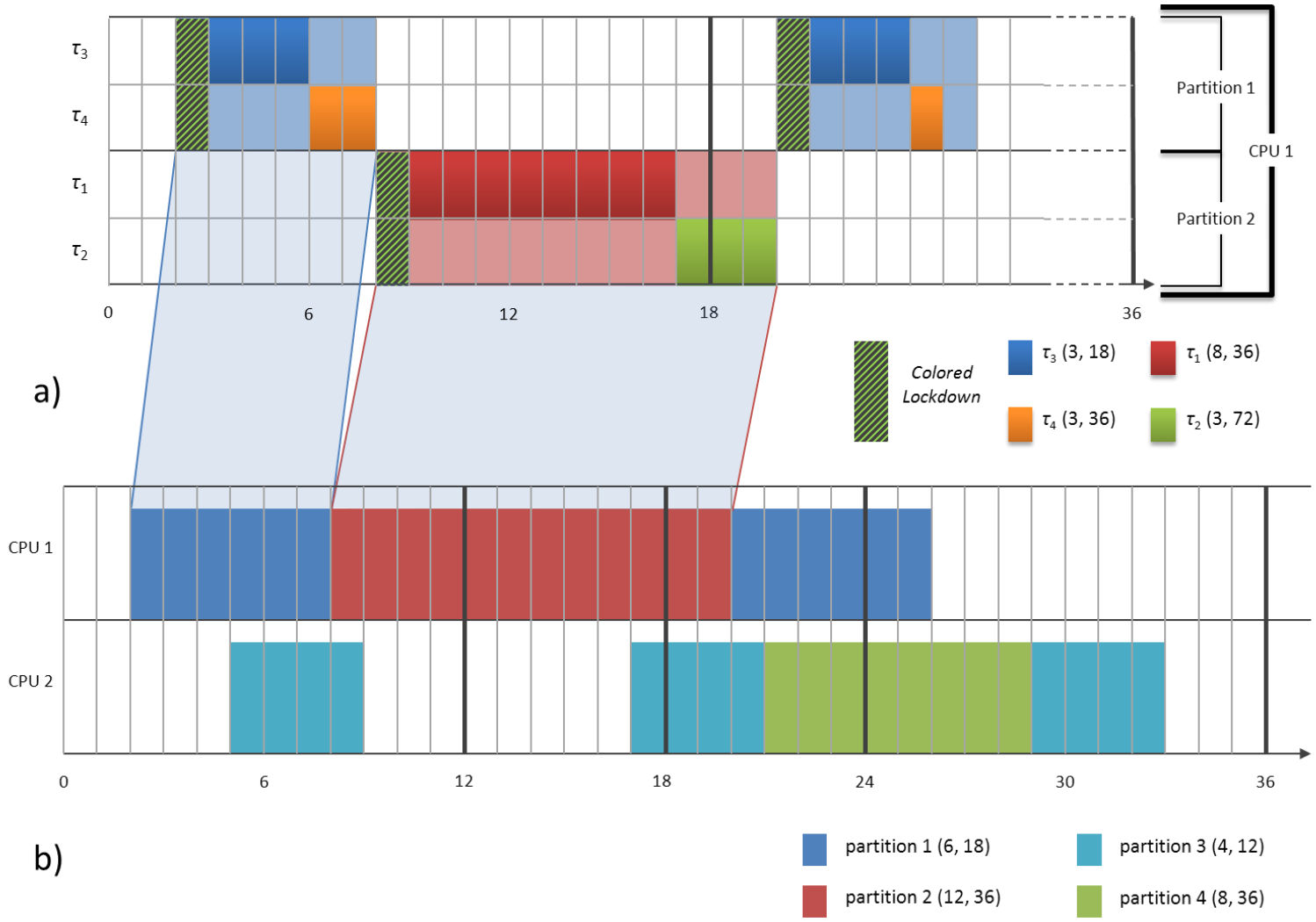
SCE: Engineering Perspective

SCE

single-core equivalence

and

IMA



SCE: Engineering Perspective

SCE

single-core equivalence

and

IMA