Green Hills SOFTWARE

Overcoming Challenges to Certifying Multicore Avionics

Brian Riley <bjriley@ghs.com>




Agenda

- Our Multicore Certification Credentials
- AMC 20-193: Use of Multi-Core Processors
- Robust Partitioning
- Bandwidth Allocation
- Cache Partitioning

Our Key Credential: World's First Multicore TSOs

CMC Electronics

PU-3000 Multicore Avionics Computer

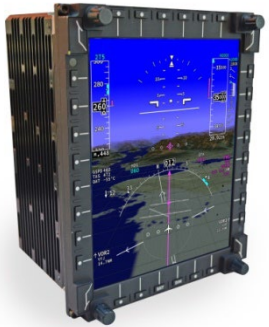
March 2021

TSO-C198 (AFGCS Equipment)

- Authorized as a Flight Director

CMC Electronics

MFD-3068 Multicore Smart Display



May 2021

3 Separate TSO Authorizations:
TSO-C113, TSO-C165, TSO-C209

- Drives up to 4 displays, incl. PFD
- Multiple apps of mixed criticality

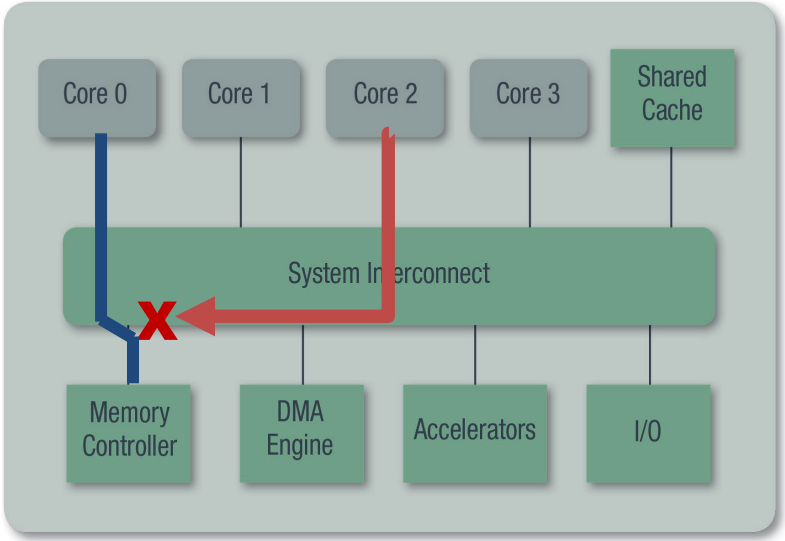
All TSOs included DO-178C DAL A and CAST-32A objectives
Both systems run the INTEGRITY®-178 tuMP™ RTOS

AMC 20-193: Use of Multi-Core Processors

- Main topic is contention for shared resources
  - Memory, cache, on-chip interconnect, and external interfaces
- 9 Objectives
  - Very little change from CAST-32A
- New Definition of Integrated Modular Avionics (IMA), which requires
  1. Robust resource partitioning
  2. Robust time partitioning

Example Shared Resource Contention:

Attempt to access DRAM while it is in use



Robust Partitioning in AMC 20-193

- **Multicore Robust Resource Partitioning is Manageable**
  - Using the memory management unit (MMU) and other well-established techniques
- **Multicore Robust Time Partitioning is Very Challenging**

> "No software partition consumes more than its allocation of execution time on the core(s) on which it executes, irrespective of whether partitions are executing on none of the other active cores or on all of the other active cores."

→ **Need a way to mitigate all or almost all sources of interference**

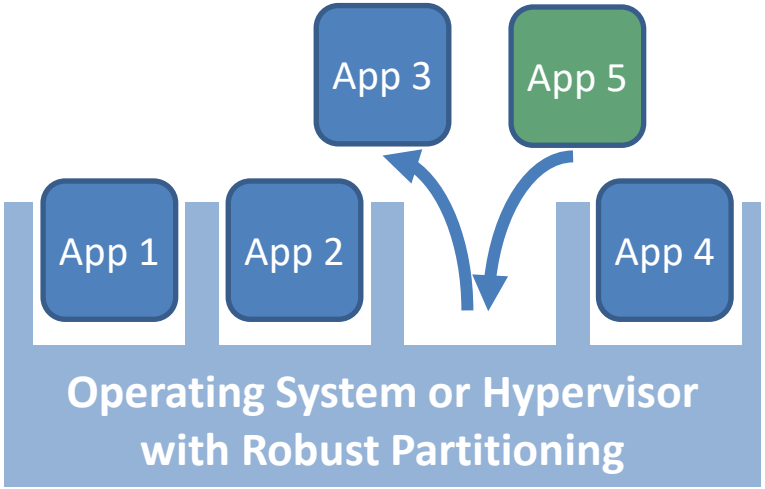
Benefits of Meeting IMA Robust Partitioning

Significantly Less Software Verification per AMC 20-193:

- Verify software applications separately
- Determine their WCETs separately

Results:

- Incremental module design and acceptance
- Applications are independently modifiable
- Platform behavior may be verified independently of specific applications

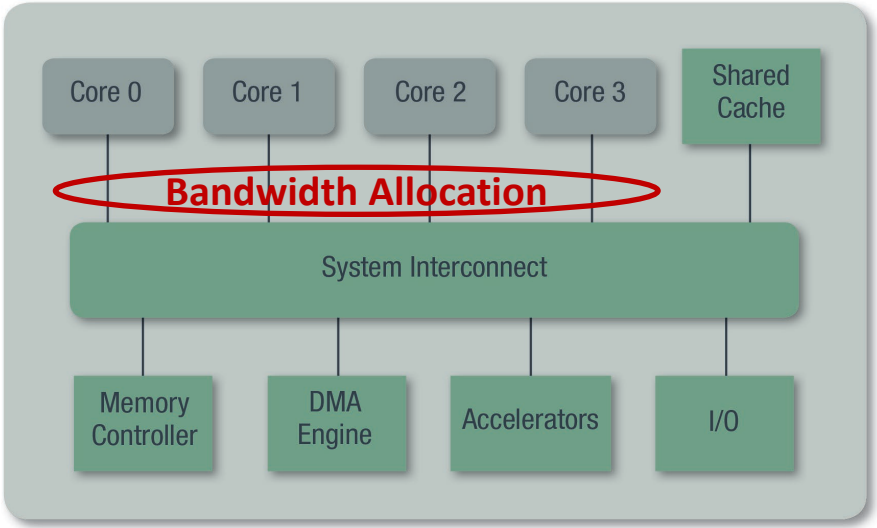


An application can be modified or replaced without testing and reverifying the entire system.

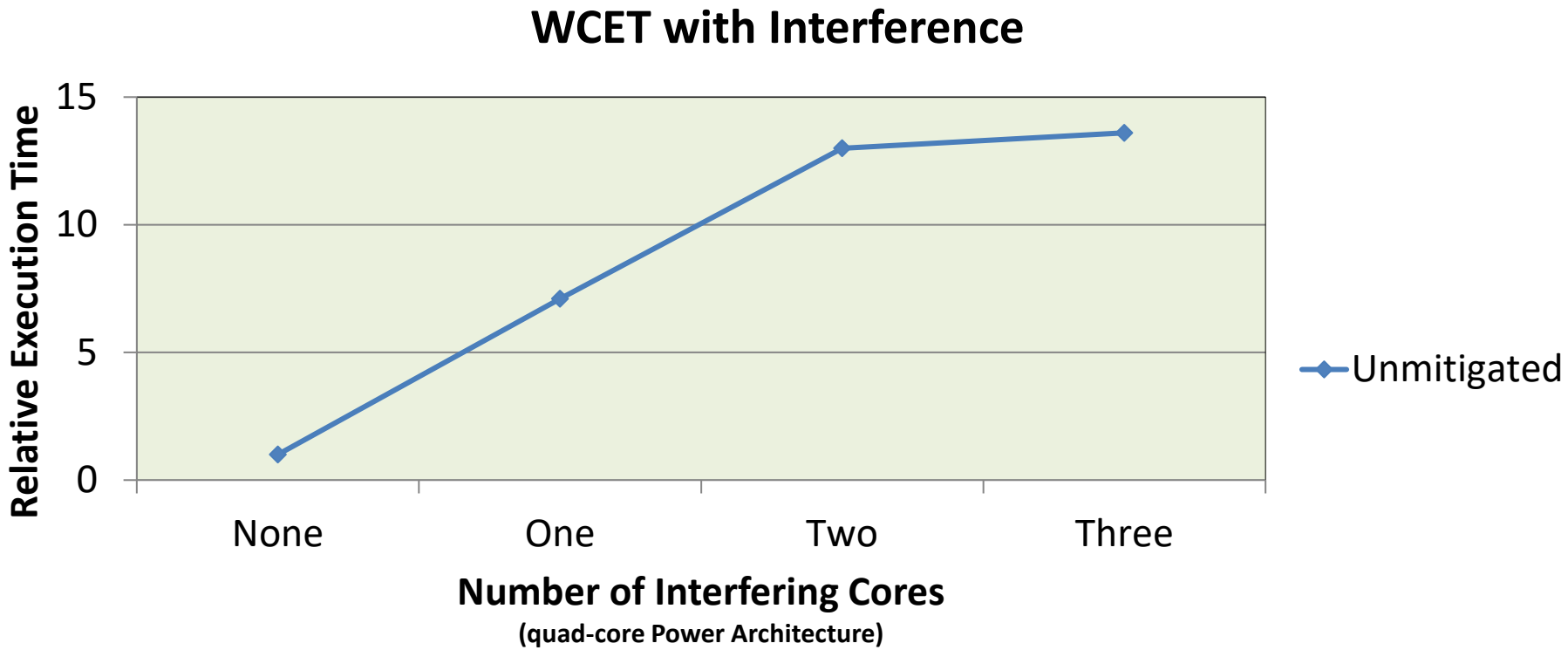
Achieving Robust Time Partitioning

General Solution: Bandwidth Allocation

- Allocate and enforce bandwidth from cores to shared resources
  - Although some CPUs have a HW implementation, it is immature, complex, and not in CPUs used for avionics
- INTEGRITY®-178 tuMP RTOS Implementation
  - Bandwidth Allocation and Monitoring (BAM)

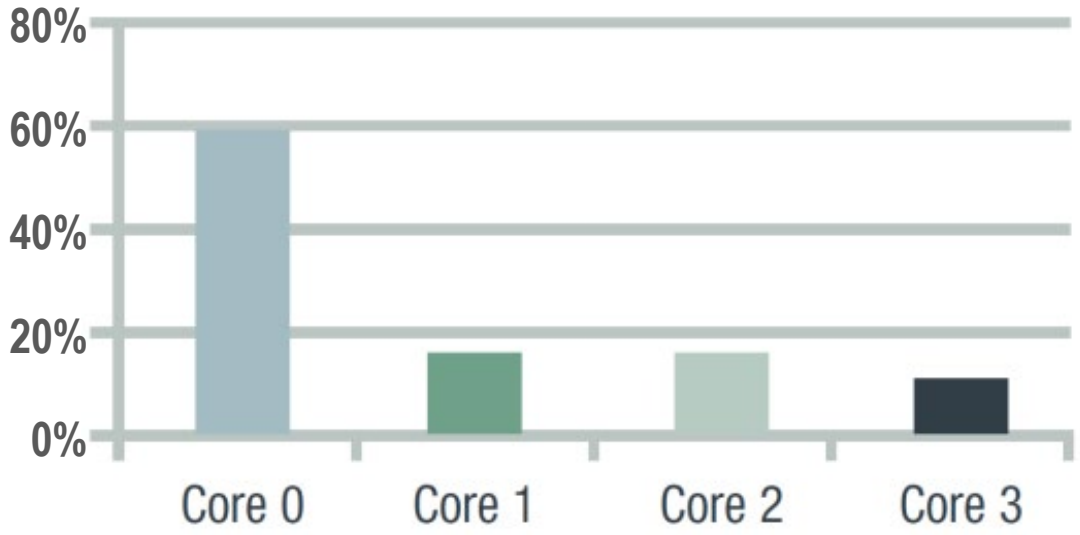


Example: Multicore DRAM Interference

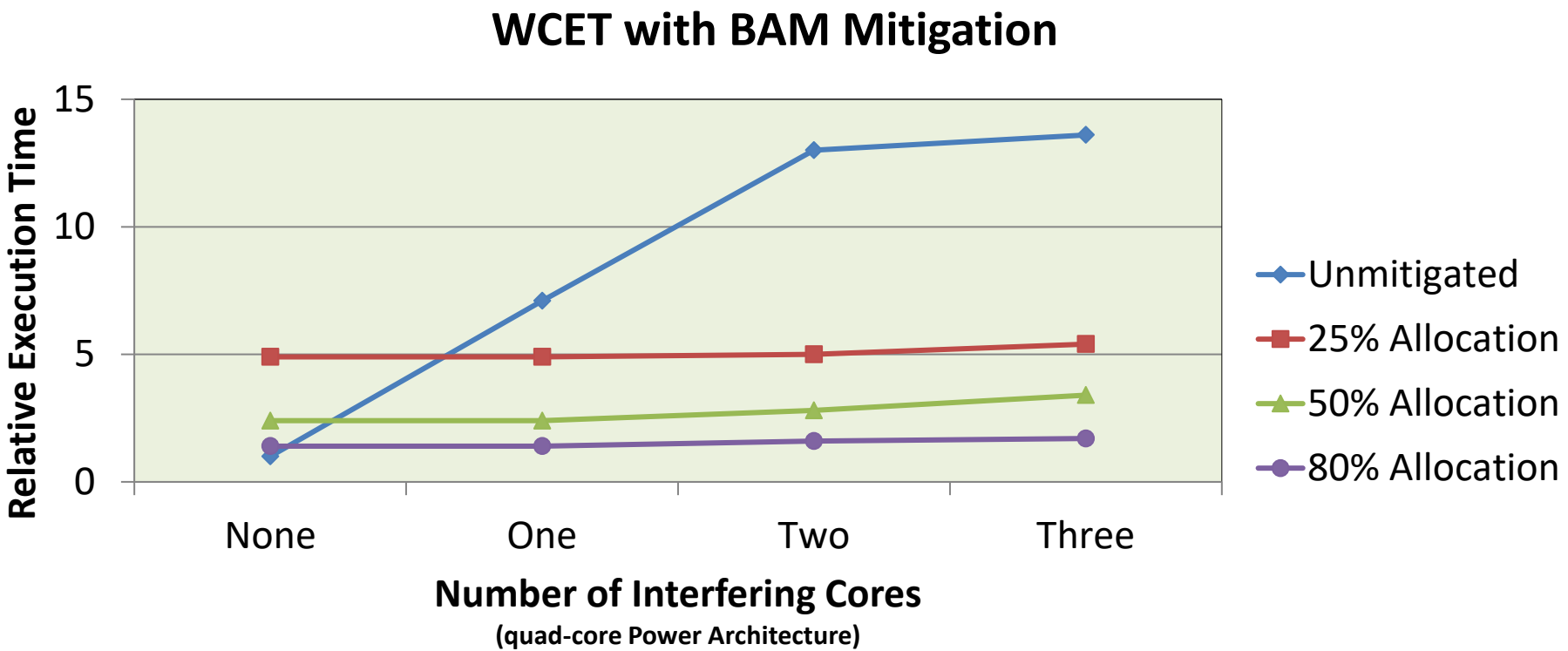


Example Bandwidth Allocations

- Software architect allocates the shared bandwidth on a per core basis
- Max bandwidth is enforced by the RTOS continuously over a very small quantum of time



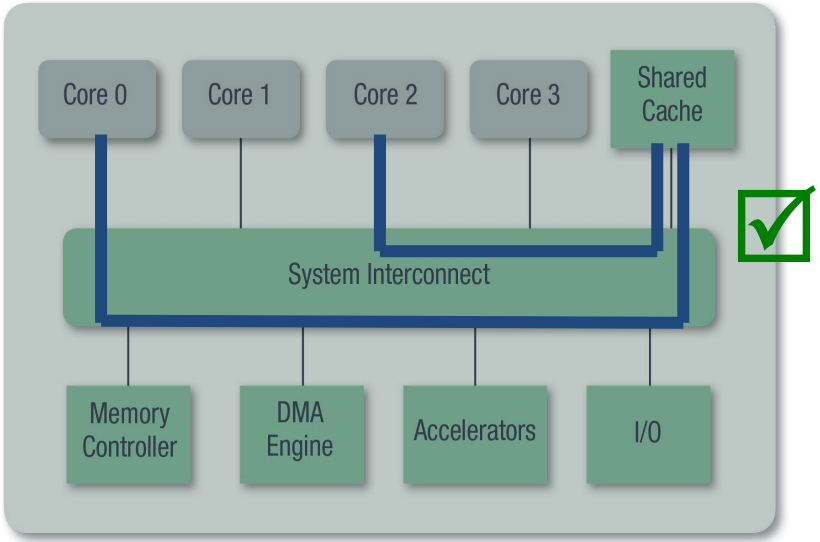
Results with BAM in INTEGRITY-178 tuMP



Shared Cache Interference is Different

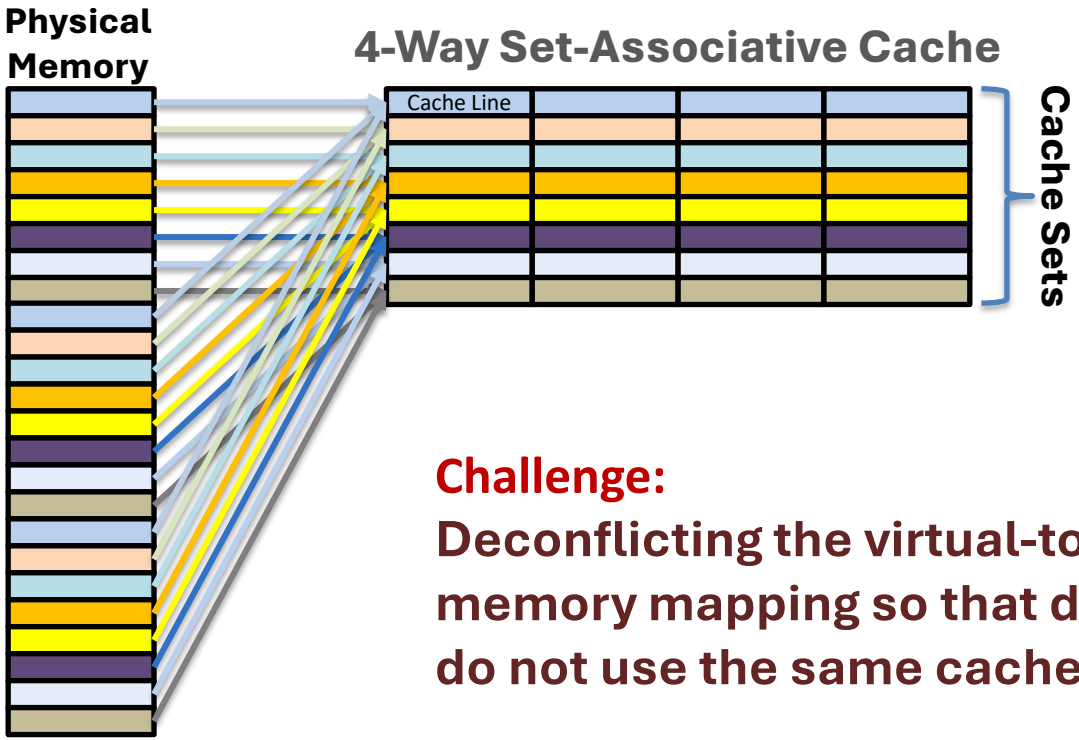
Contention for Access in Not an Issue

Shared Cache Supports Simultaneous Access



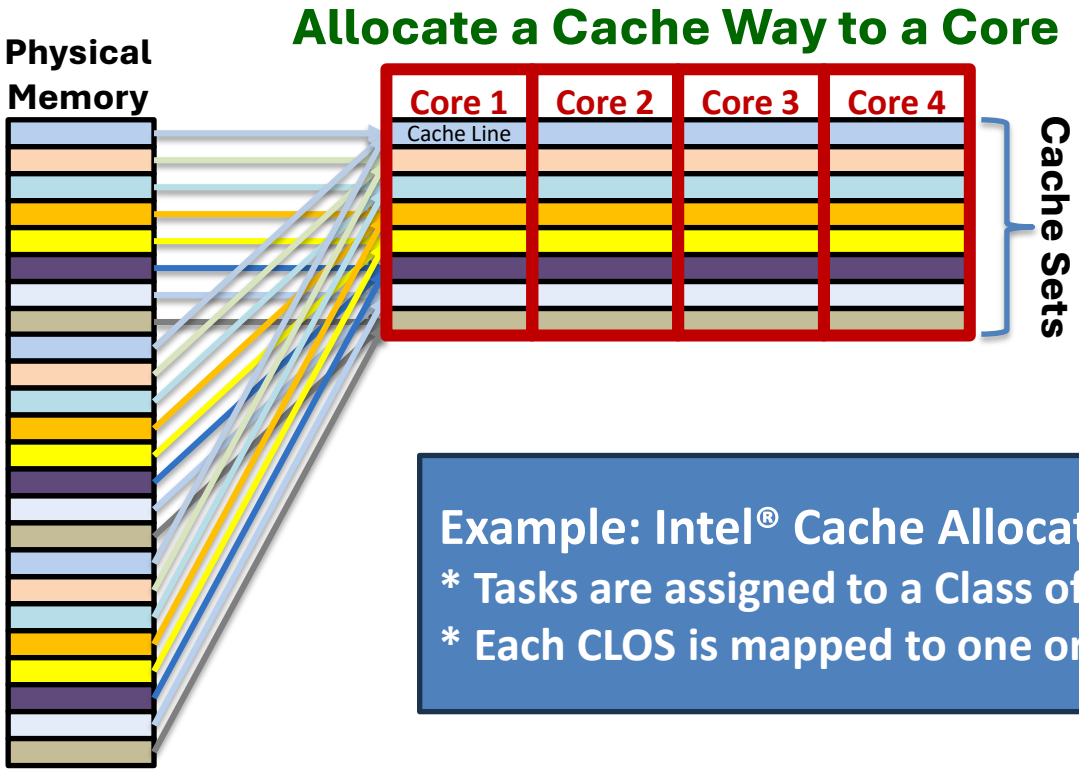
- **Problem is Inter-Core Cache Line Eviction**
  - **Task on Core 0 goes to read its data, but a task on Core 2 caused new data to be loaded instead**

Multiple Memory Locations Map to Same Cache Line/Set



Challenge:
Deconflicting the virtual-to-physical memory mapping so that different cores do not use the same cache line

Hardware-Based Cache Partitioning to the Rescue!



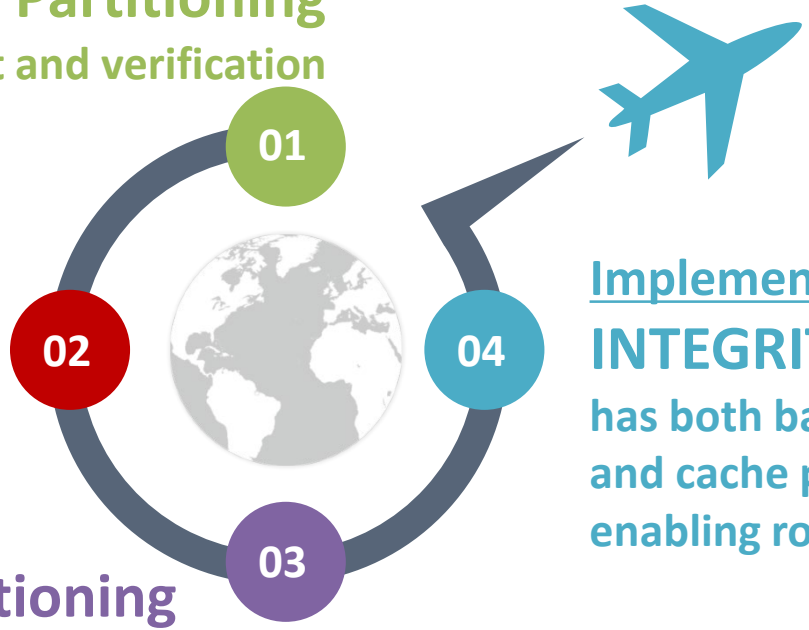
Example: Intel® Cache Allocation Technology

* Tasks are assigned to a Class of Service (CLOS)
* Each CLOS is mapped to one or more cache ways

Overcoming Multicore Certification Challenges

Goal: Robust Partitioning

eases development and verification



Solution:

Bandwidth Allocation

mitigates most sources of multicore interference

Cache Partitioning

mitigates non-contention interference

Implementation:

INTEGRITY-178 tuMP

has both bandwidth allocation and cache partitioning thus enabling robust partitioning