WCET(*m*) Estimation in Multi-Core Systems using SCE

SCE
single-core equivalence

University of Illinois at Urbana-Champaign

Renato Mancuso

Marco Caccamo

Lui Sha

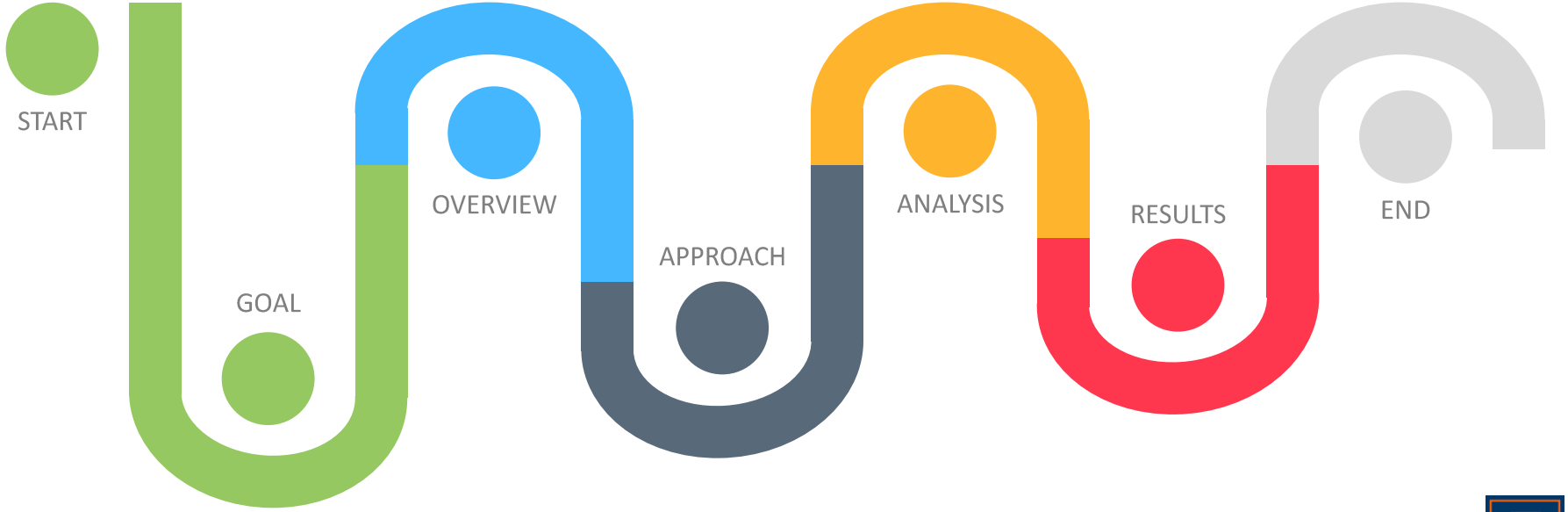
Rodolfo Pellizzoni

Heechul Yun

KU THE UNIVERSITY OF KANSAS

Road Map

presentation content



Real-Time & Multi-Core

the upcoming migration

benefits

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

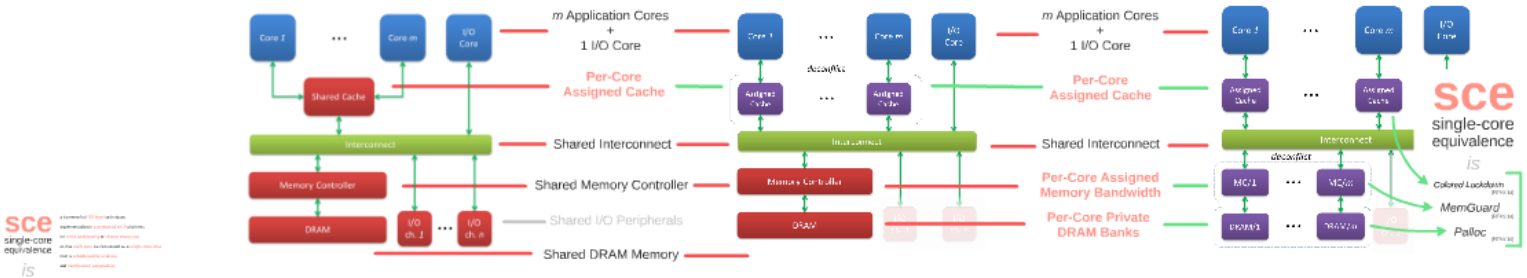
industry has a **large body** of certified **single-core** hard real-time **software**

old software **migrated en-masse**, **new software** developed to use extra CPU

this is not **risk-free**

Single-Core Equivalence

practical multi-core migration



sce

single-core equivalence

is

a framework of **OS-level** techniques

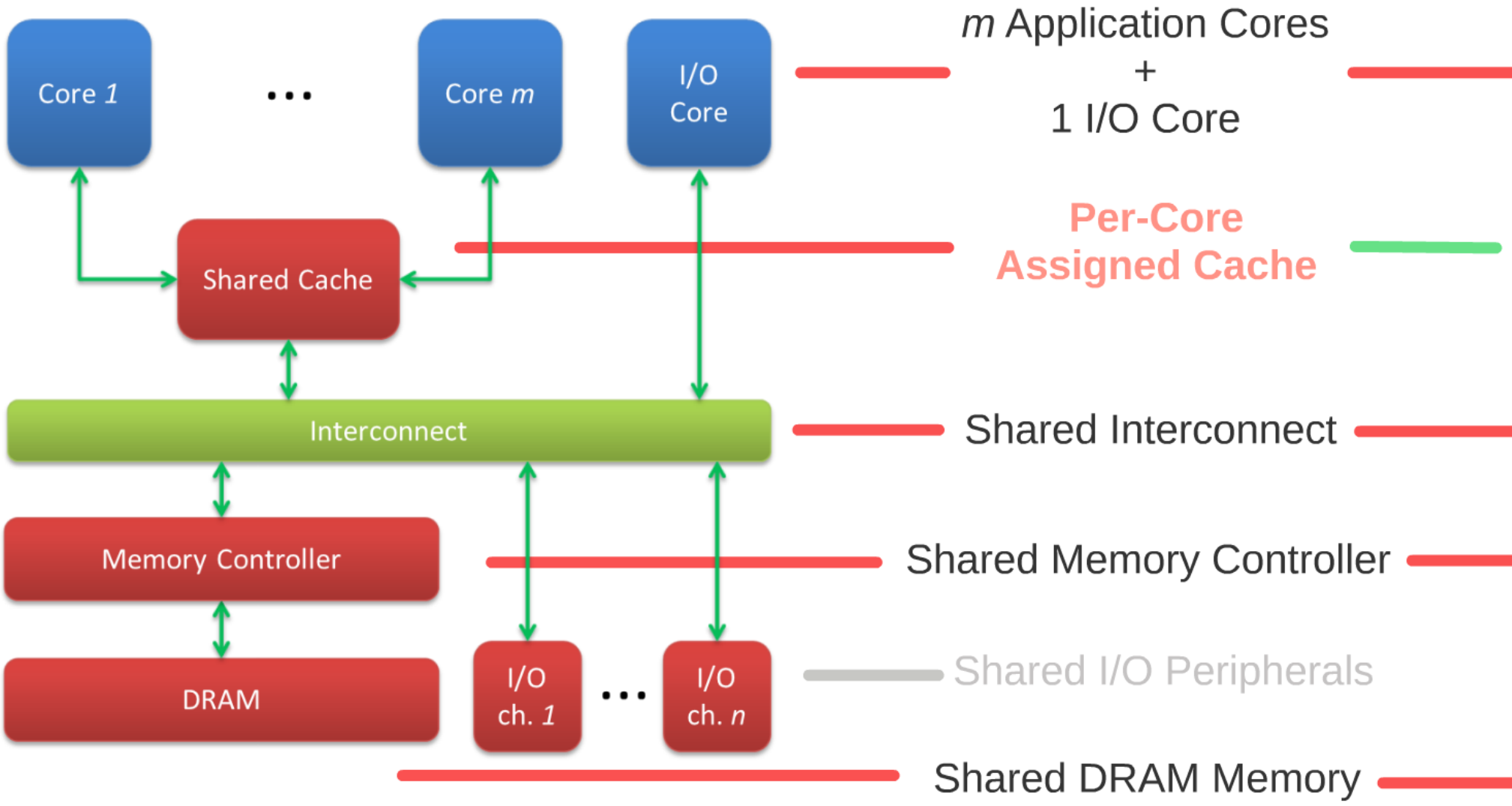
implementable on **commercial MCP** platforms

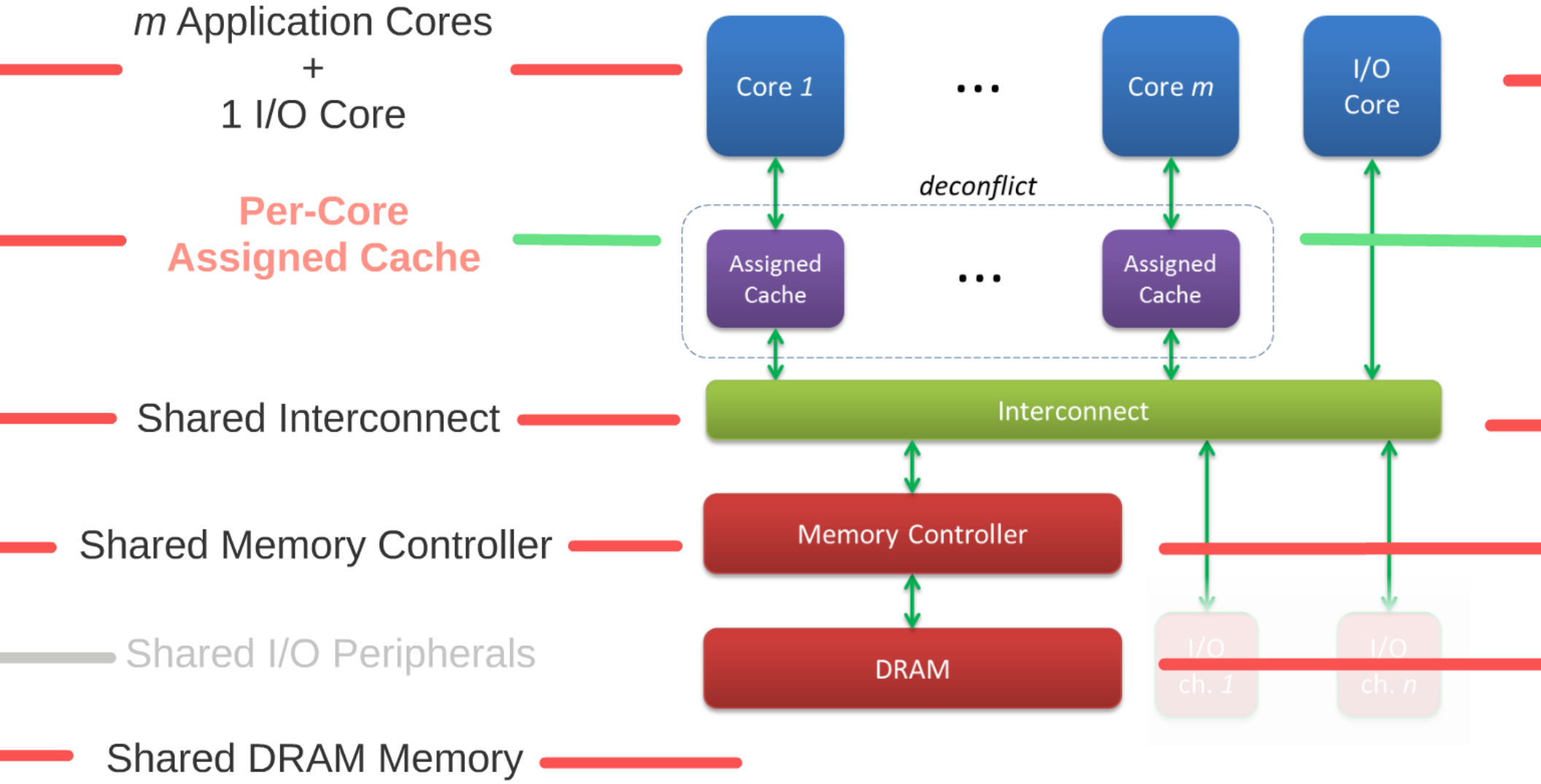
for **strict partitioning** of **shared resources**

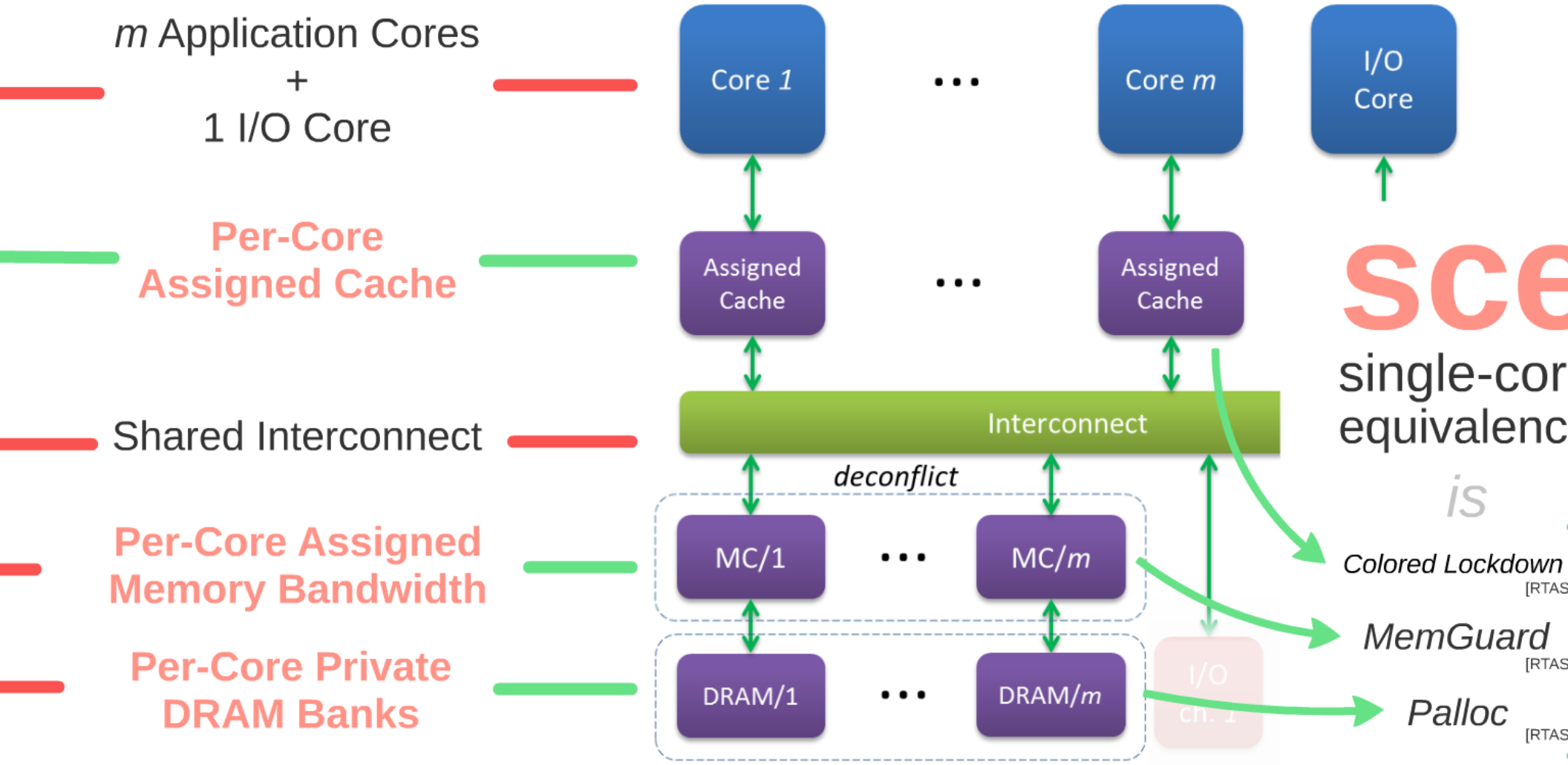
so that **each core** can be treated as a **single-core chip**

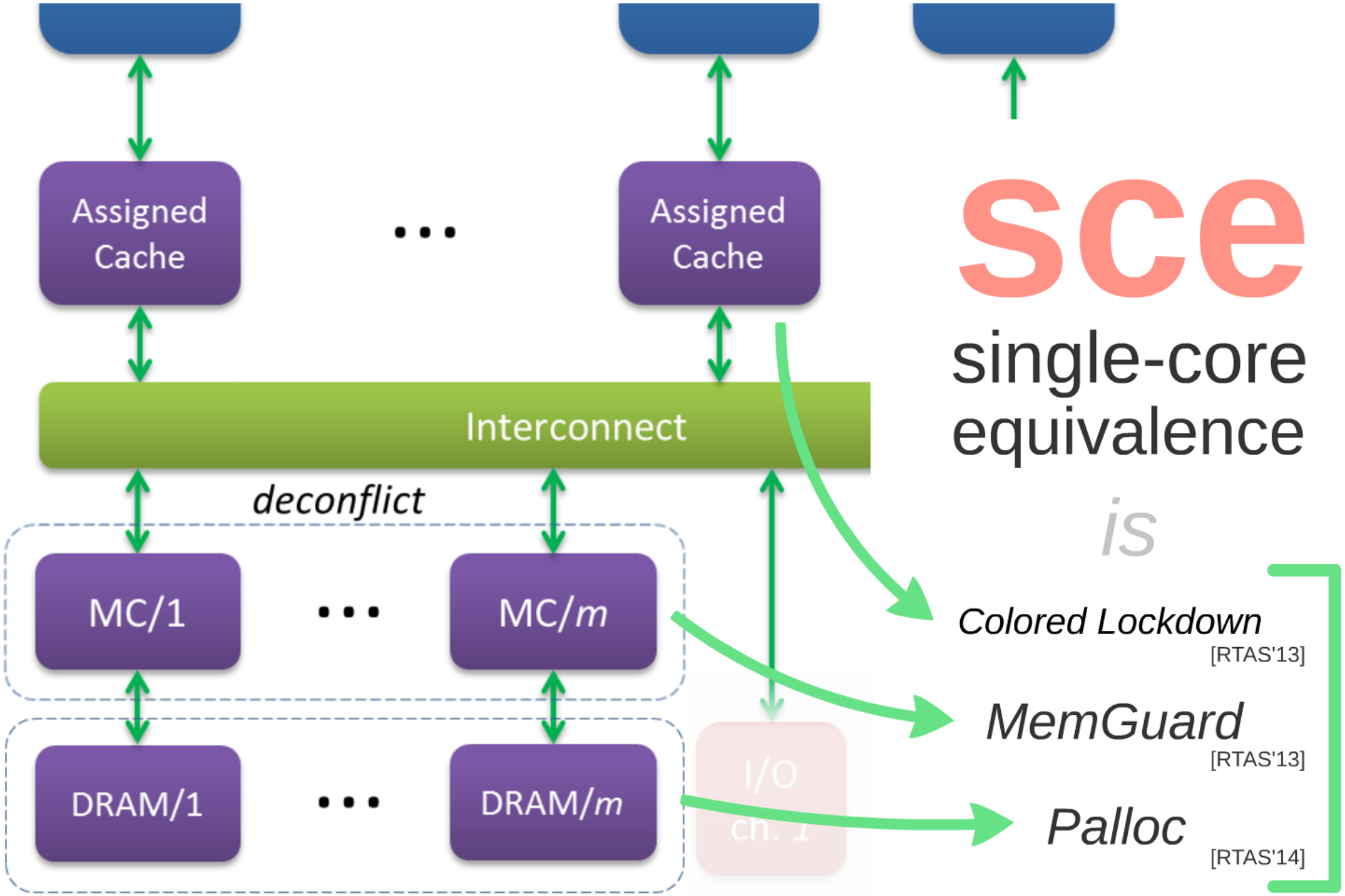
from a **schedulability analysis**

and **certification perspective**


Core 1
Core m
I/O Core
m Application Cores
+
1 I/O Core
Shared Cache
Per-Core Assigned Cache
Interconnect
Shared Interconnect
Memory Controller
Shared Memory Controller
DRAM
I/O ch. 1
I/O ch. n
Shared I/O Peripherals
Shared DRAM Memory


m Application Cores
+
1 I/O Core
Per-Core Assigned Cache
Shared Interconnect
Shared Memory Controller
Shared I/O Peripherals
Shared DRAM Memory
Core *1*
Core *m*
I/O Core
deconflict
Assigned Cache
Assigned Cache
Interconnect
Memory Controller
DRAM
I/O ch. *1*
I/O ch. *n*

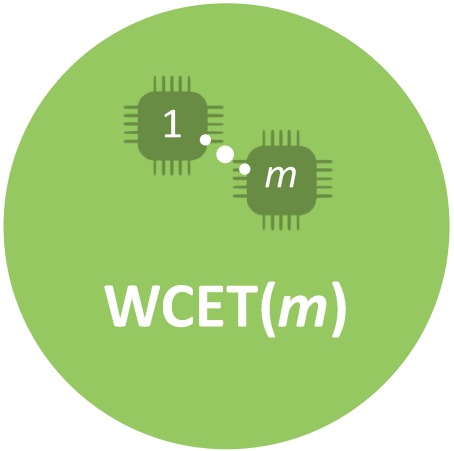

m Application Cores
+
1 I/O Core
Per-Core
Assigned Cache
Shared Interconnect
Per-Core Assigned
Memory Bandwidth
Per-Core Private
DRAM Banks
Core 1
Core m
I/O
Core
Assigned
Cache
Assigned
Cache
Interconnect
deconflict
MC/1
MC/m
DRAM/1
DRAM/m
I/O
ch. 1
sce
single-cor
equivalenc
is
Colored Lockdown
[RTAS
MemGuard
[RTAS
Palloc
[RTAS


Assigned Cache
Assigned Cache
Interconnect
deconflict
MC/1
MC/m
DRAM/1
DRAM/m
I/O ch. 1
sce
single-core equivalence
is
Colored Lockdown
[RTAS'13]
MemGuard
[RTAS'13]
Palloc
[RTAS'14]

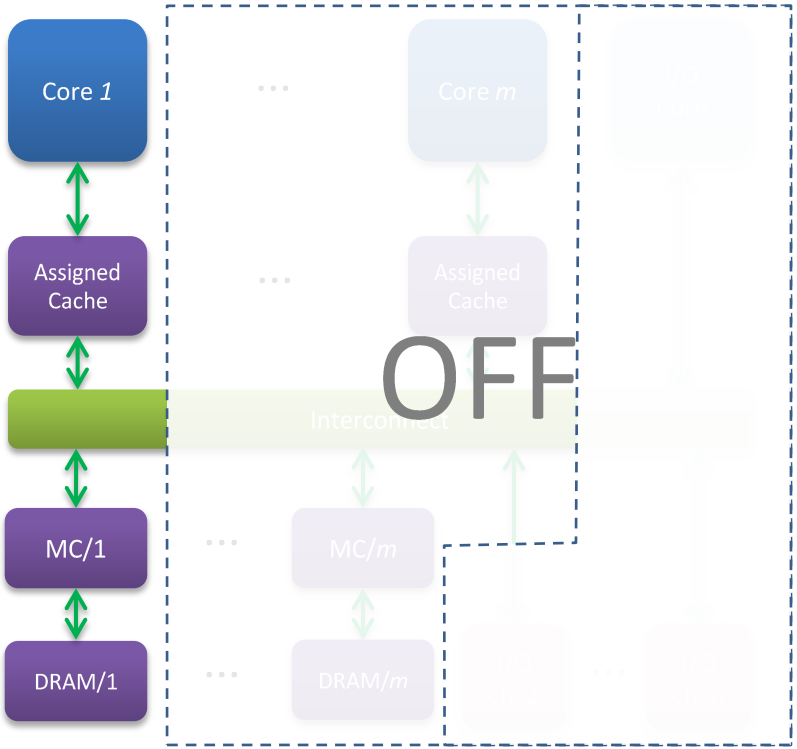
Modular Certification

certify core-by-core, up to m

Certification almost identical to single-core case

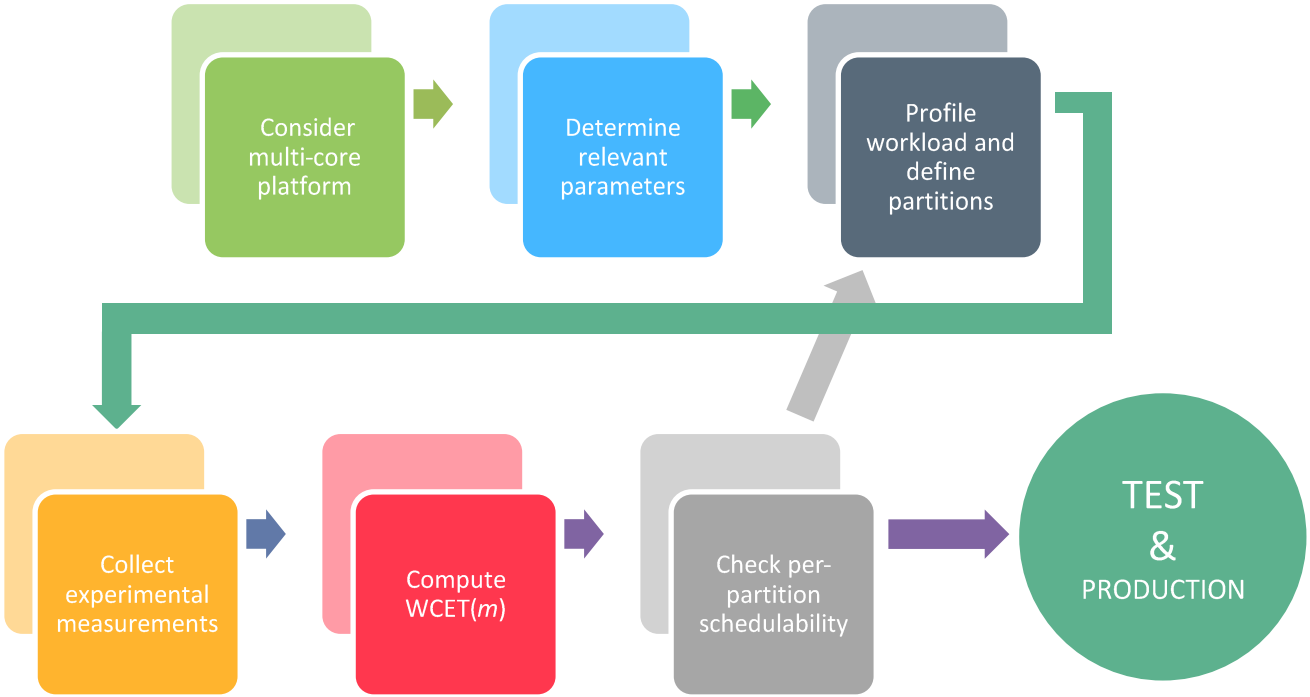


Certification done for *m* or **less** active cores



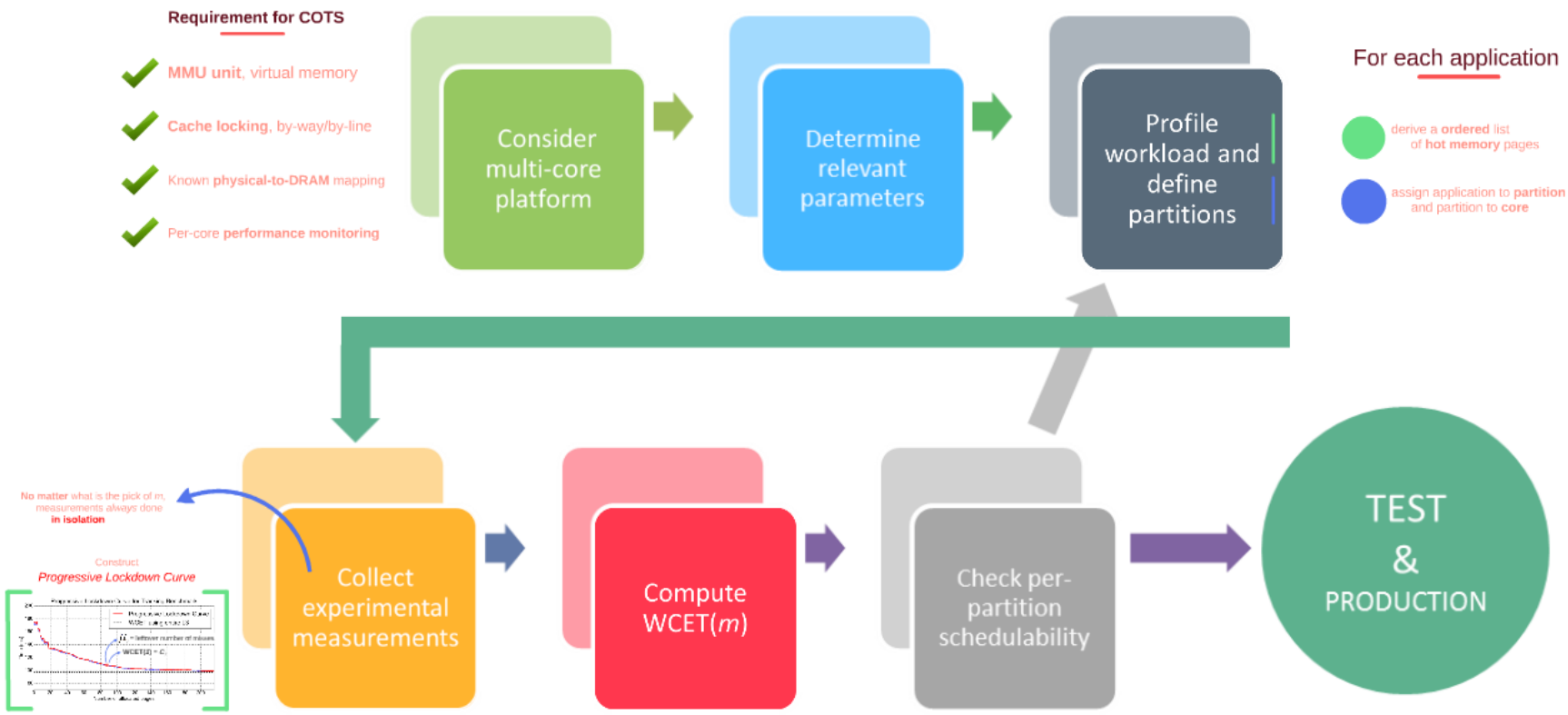
SCE Methodology

a step-by-step design



SCE Methodology

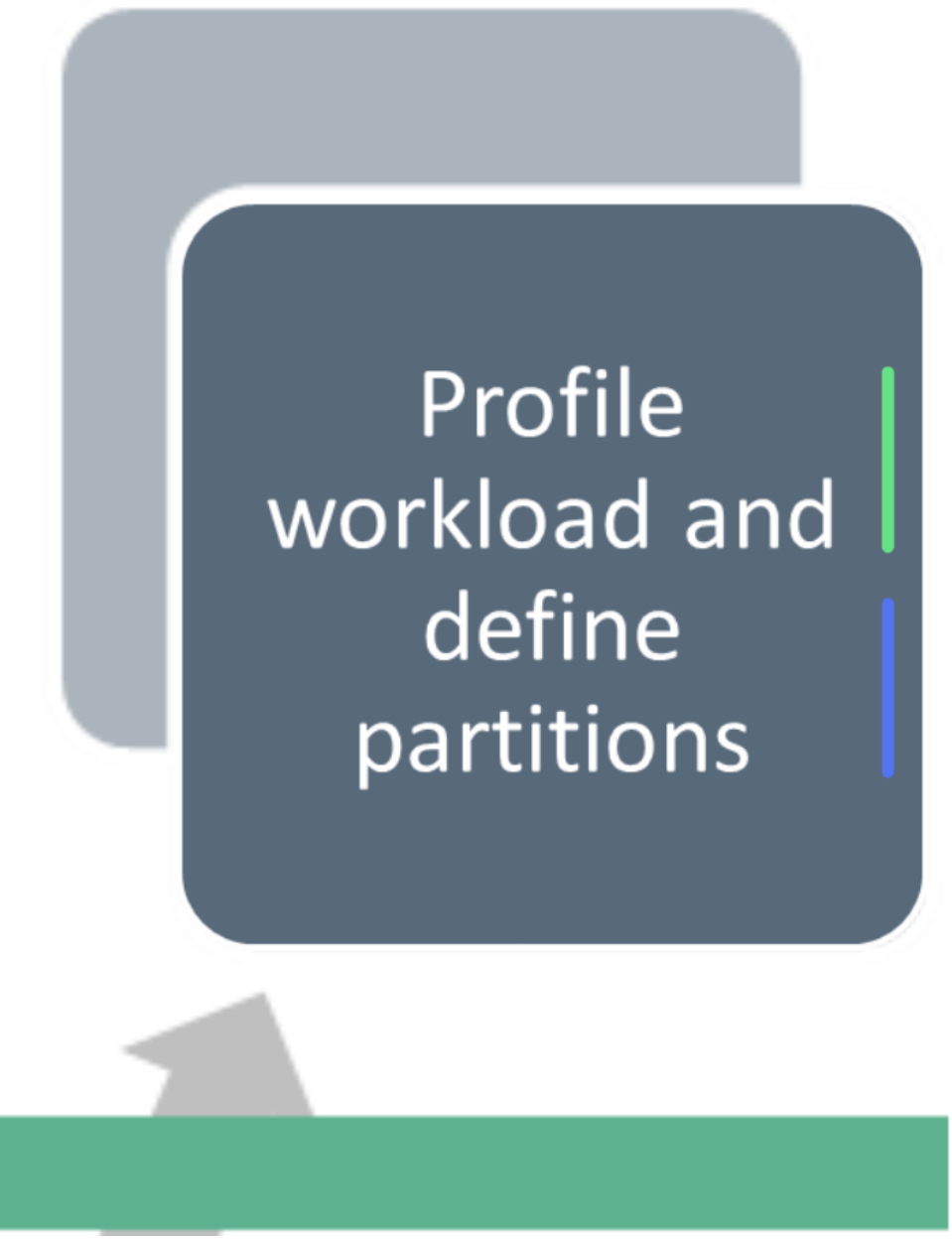
a step-by-step design



Requirement for COTS

- **MMU unit**, virtual memory
- **Cache locking**, by-way/by-line
- Known **physical-to-DRAM** mapping
- Per-core **performance monitoring**

Consider multi-core platform

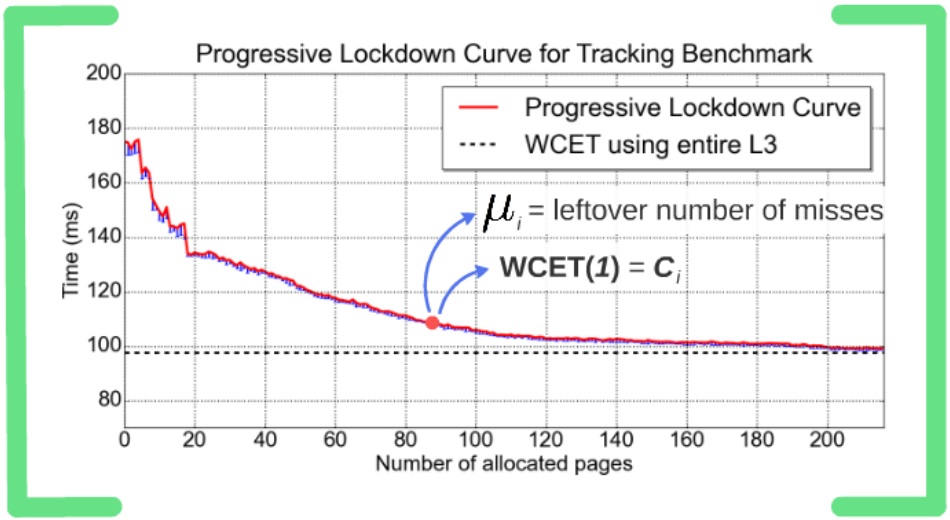


For each application

- derive a **ordered** list of **hot memory** pages
- assign application to **partition** and partition to **core**

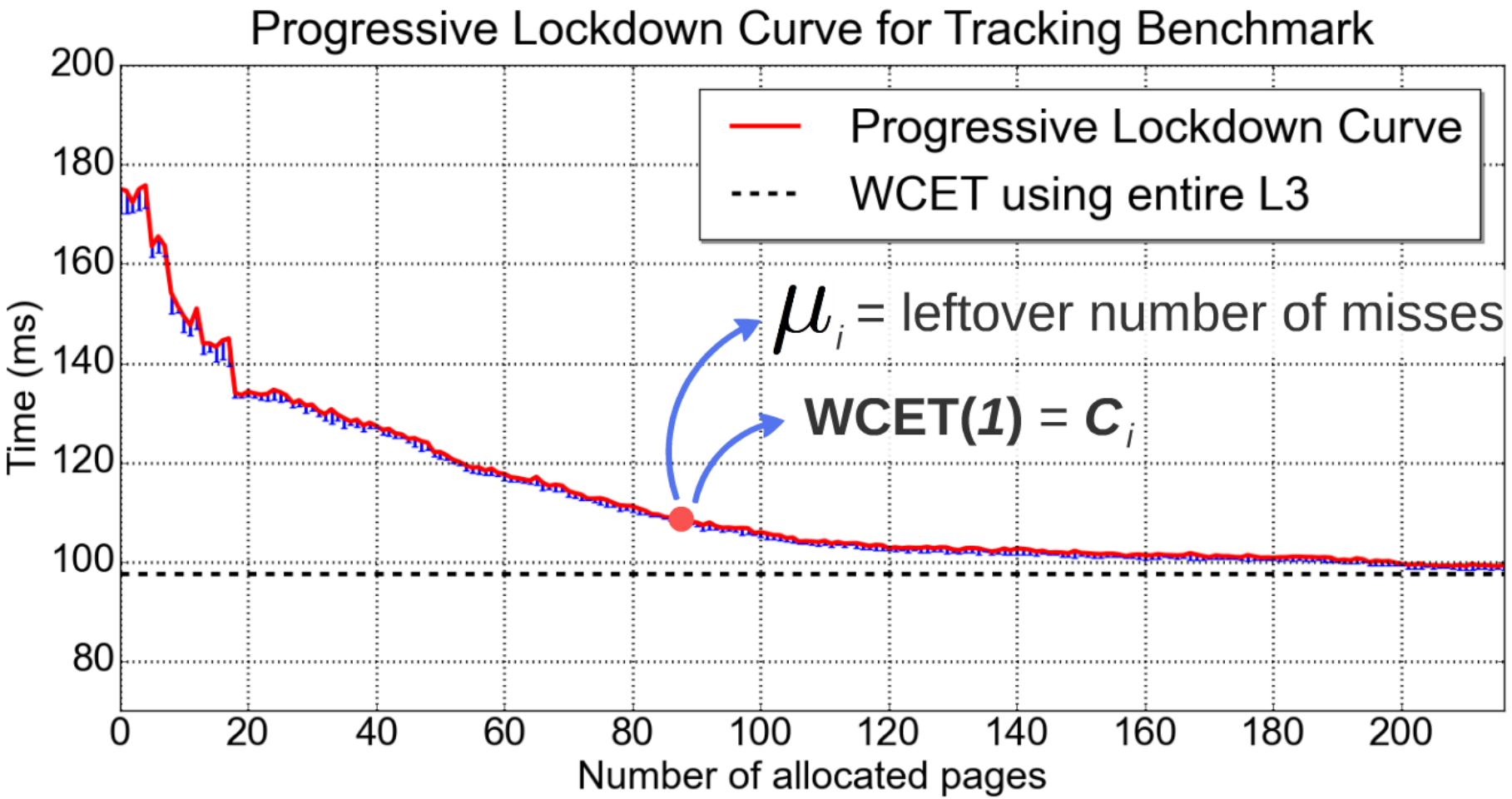
No matter what is the pick of *m*,
measurements *always* done
in isolation

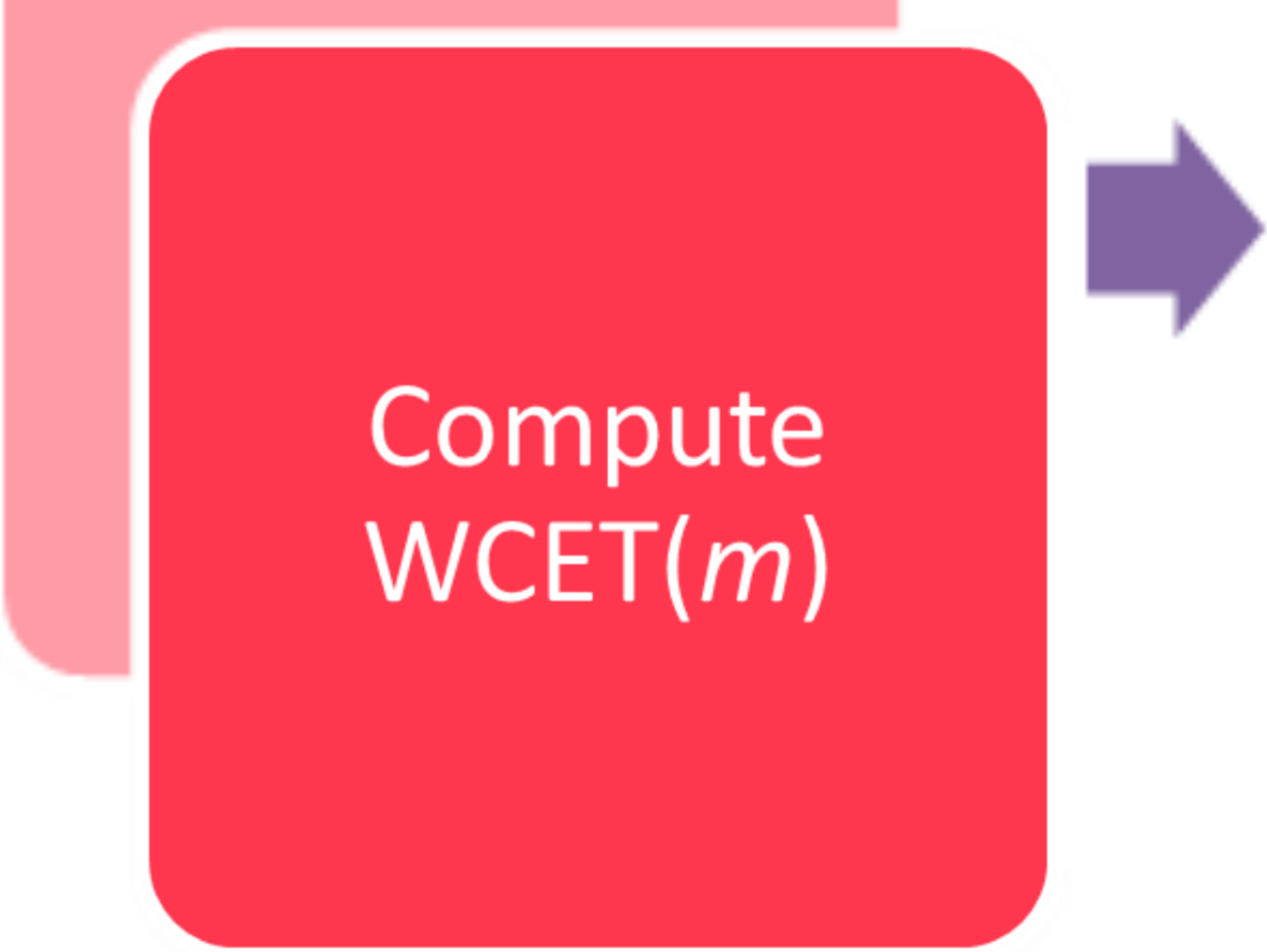
Construct
Progressive Lockdown Curve



Collect experimental measurements

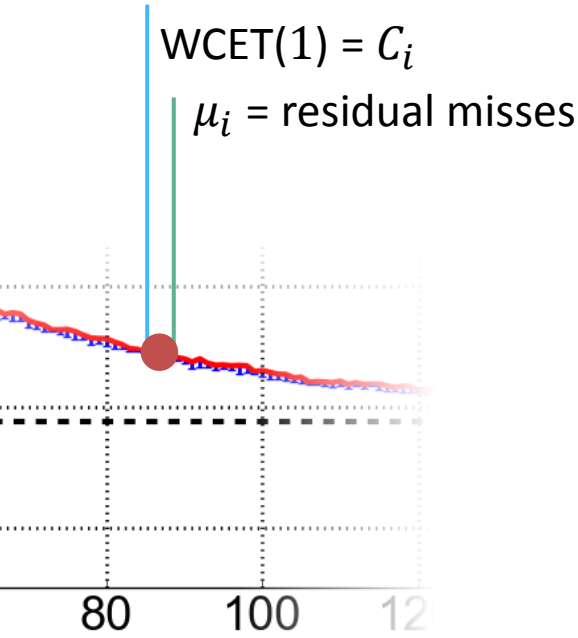
Progressive Lockdown Curve




Compute
WCET(*m*)

WCET(m) Calculation

accounting for resource partitioning



$$\text{WCET}(m)_i$$

$$C_i + \mu_i \cdot L_{size}\left(\frac{m}{BW_{min}} - \frac{1}{BW_{max}}\right)$$

$$\text{WCET}(m)_i$$

$$C_i + \mu_i \cdot L_{size}\left(\frac{m}{BW_{min}} - \frac{1}{BW_{max}}\right)$$

Response Time

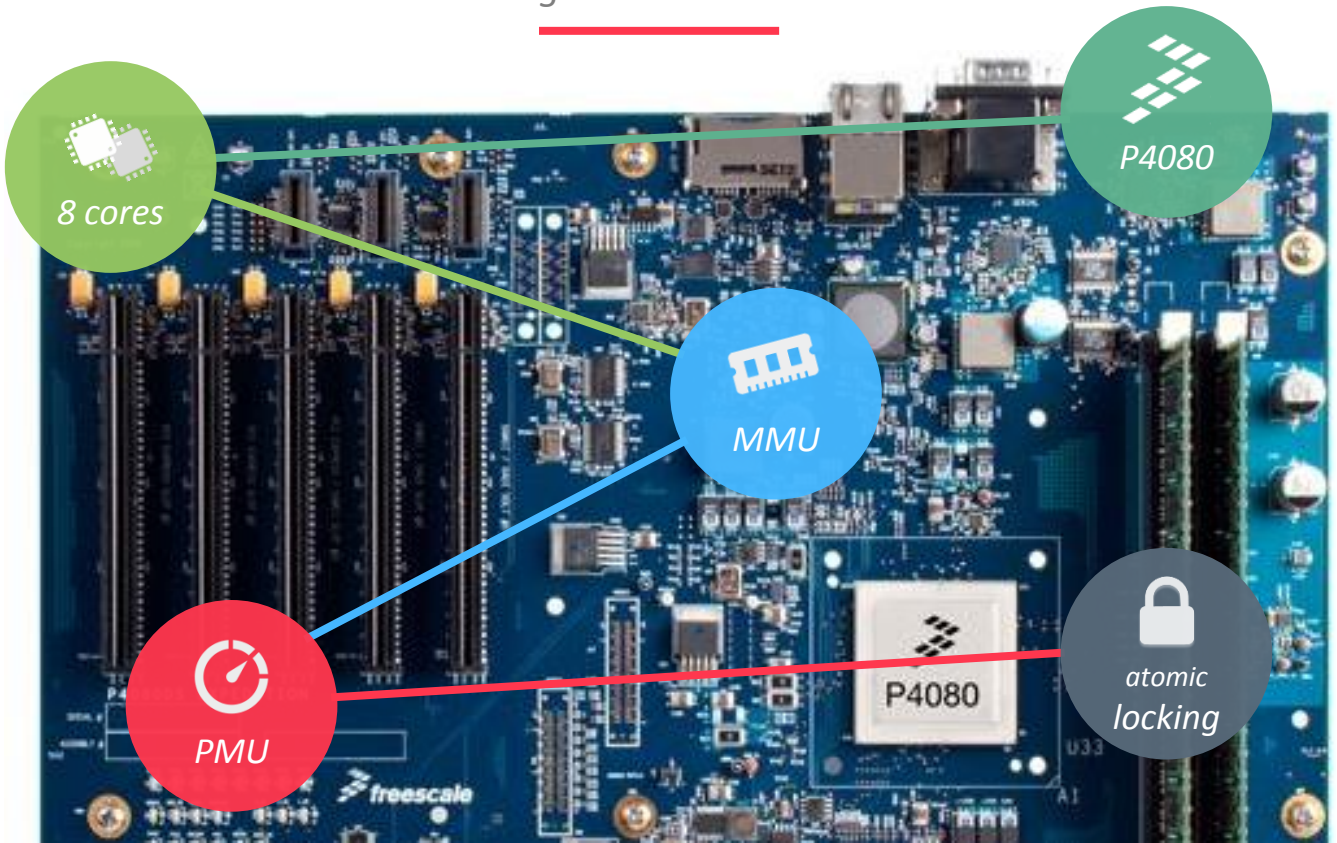
reusing a well-known approach

$$R_i(k+1) =$$

$$\text{WCET}(m)_i + \sum_{\tau_j \in hp(i)} \left\lceil \frac{R_i(k)}{T_j} \right\rceil \text{WCET}(m)_j + B_{MG}$$

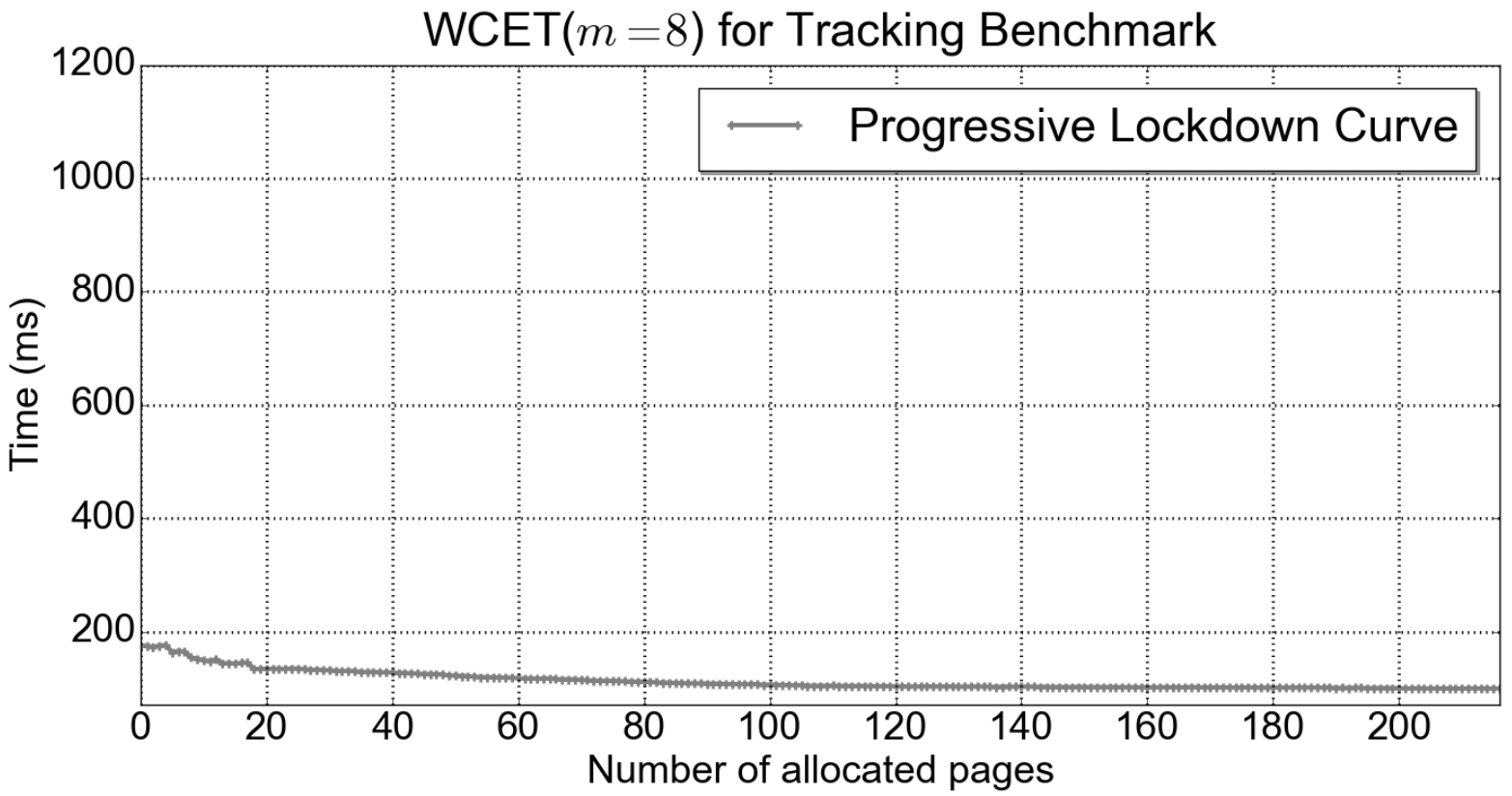
SCE Validation

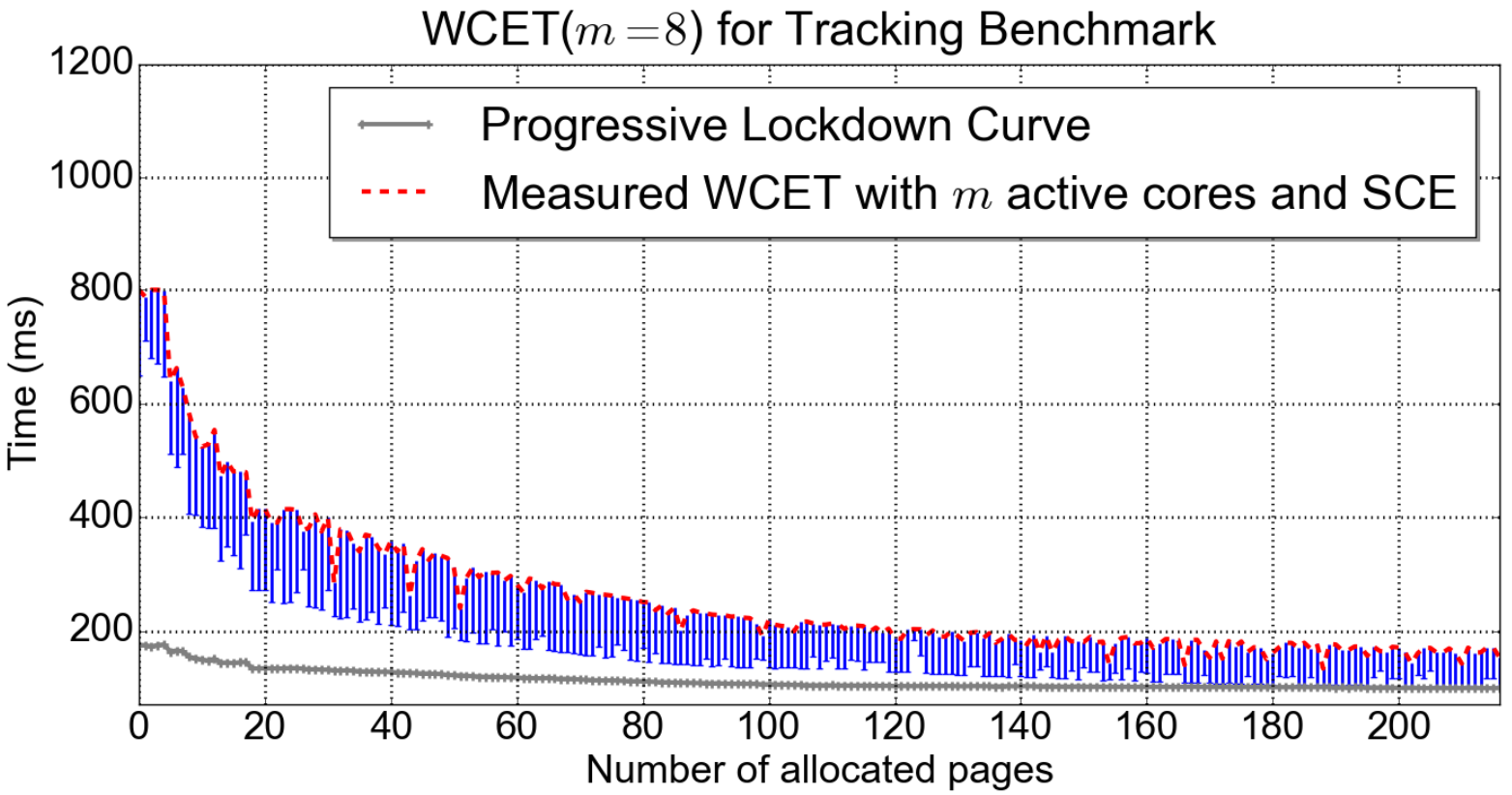
using COTS hardware

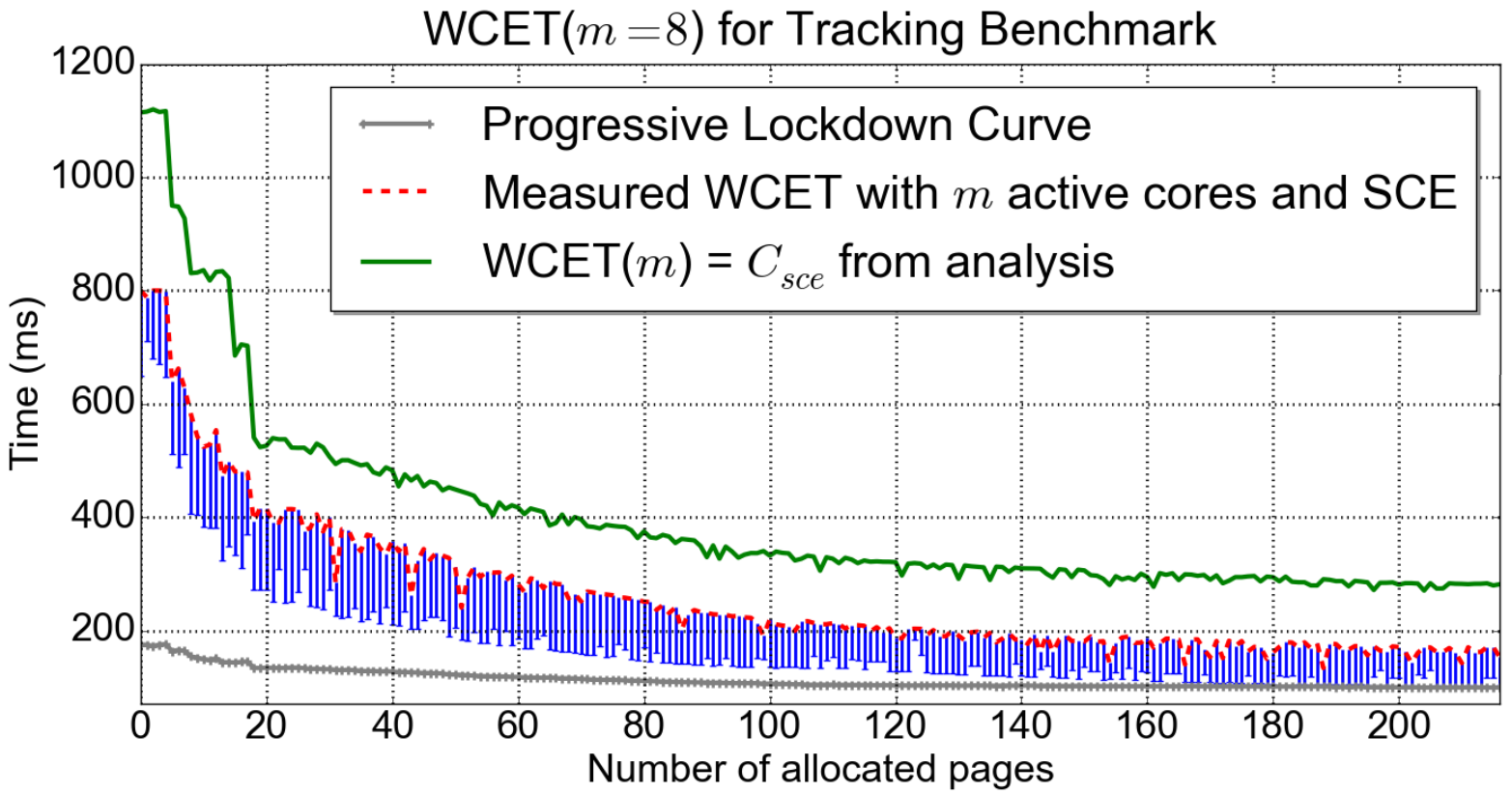


SCE Validation

WCET(m) bound


WCET($m = 8$) for Tracking Benchmark
Progressive Lockdown Curve
Time (ms)
1200
1000
800
600
400
200
0
20
40
60
80
100
120
140
160
180
200
Number of allocated pages


WCET($m = 8$) for Tracking Benchmark
Progressive Lockdown Curve
Measured WCET with m active cores and SCE
Time (ms)
Number of allocated pages
1200
1000
800
600
400
200
0
20
40
60
80
100
120
140
160
180
200


WCET($m = 8$) for Tracking Benchmark
Progressive Lockdown Curve
Measured WCET with m active cores and SCE
WCET(m) = C_{sce} from analysis
Time (ms)
Number of allocated pages
1200
1000
800
600
400
200
0
20
40
60
80
100
120
140
160
180
200

SCE Validation

WCET(m) sensitivity

$$\text{WCET}(m)_i$$

$$C_i + \mu_i \cdot L_{size} \left(\frac{m}{BW_{min}} - \frac{1}{BW_{max}} \right)$$

SCE Validation

WCET(m) sensitivity

$$\text{WCET}(m)_i$$

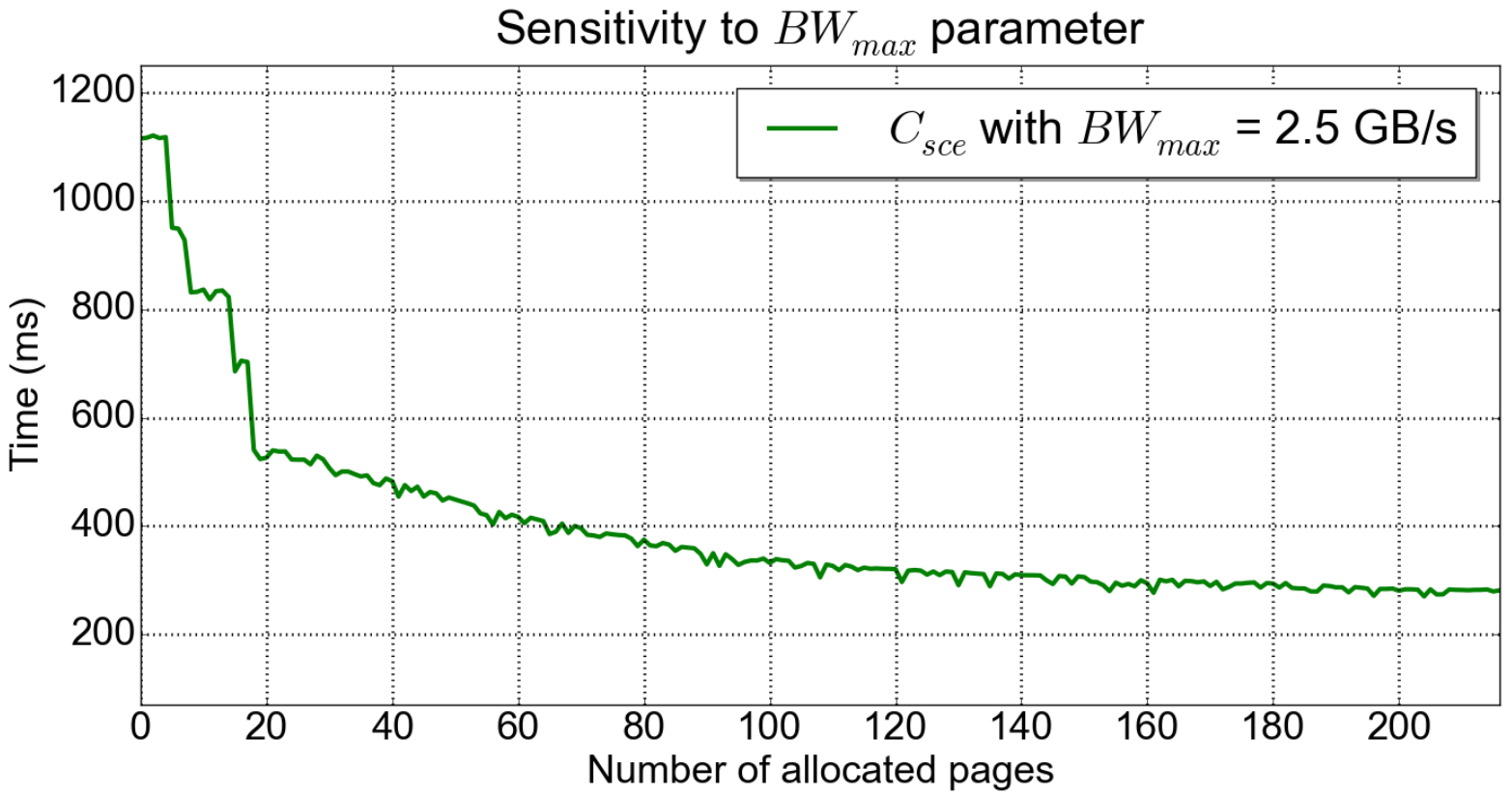
$$C_i + \mu_i \cdot L_{size} \left(\frac{m}{BW_{min}} - \frac{1}{BW_{max}} \right)$$

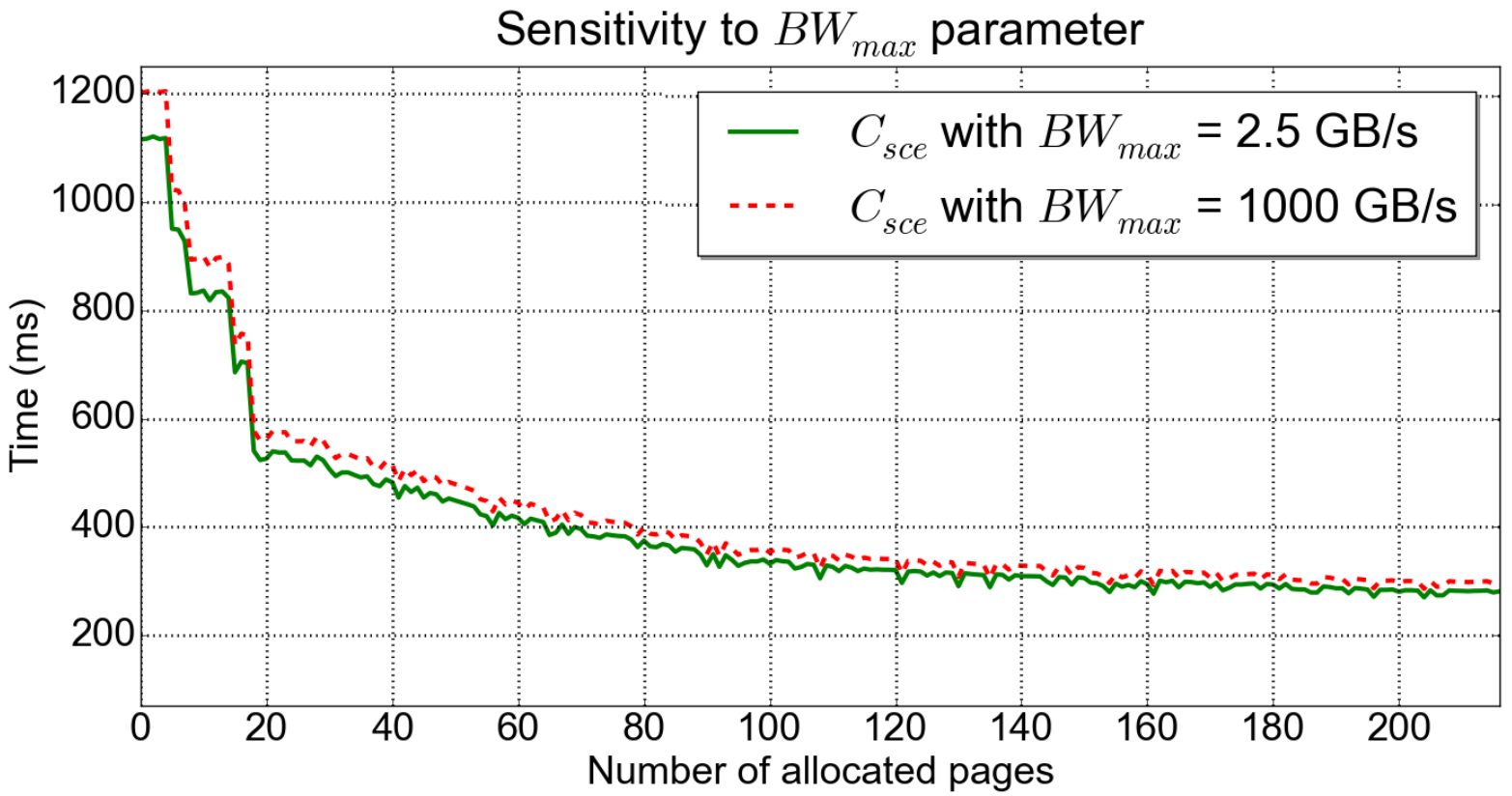
SCE Validation

WCET(m) sensitivity

$$\text{WCET}(m)_i$$

$$C_i + \mu_i \cdot L_{size}\left(\frac{m}{BW_{min}}\right)$$


Sensitivity to BW_{max} parameter
C_{sce} with BW_{max} = 2.5 GB/s
Time (ms)
1200
1000
800
600
400
200
0
20
40
60
80
100
120
140
160
180
200
Number of allocated pages


Sensitivity to BW_{max} parameter
C_{sce} with BW_{max} = 2.5 GB/s
C_{sce} with BW_{max} = 1000 GB/s
Time (ms)
1200
1000
800
600
400
200
0
20
40
60
80
100
120
140
160
180
200
Number of allocated pages

Summary

1. **Select *m*, SCE** provides a constant **WCET(*m*)**
2. Resource partitioning **at OS-level,** use on **COTS**
3. Reuse **single-core software** & **practices**

Thanks.

University of Illinois at Urbana-Champaign

KU
THE UNIVERSITY OF KANSAS