



Resource-Aware Replication on Heterogeneous Multicores

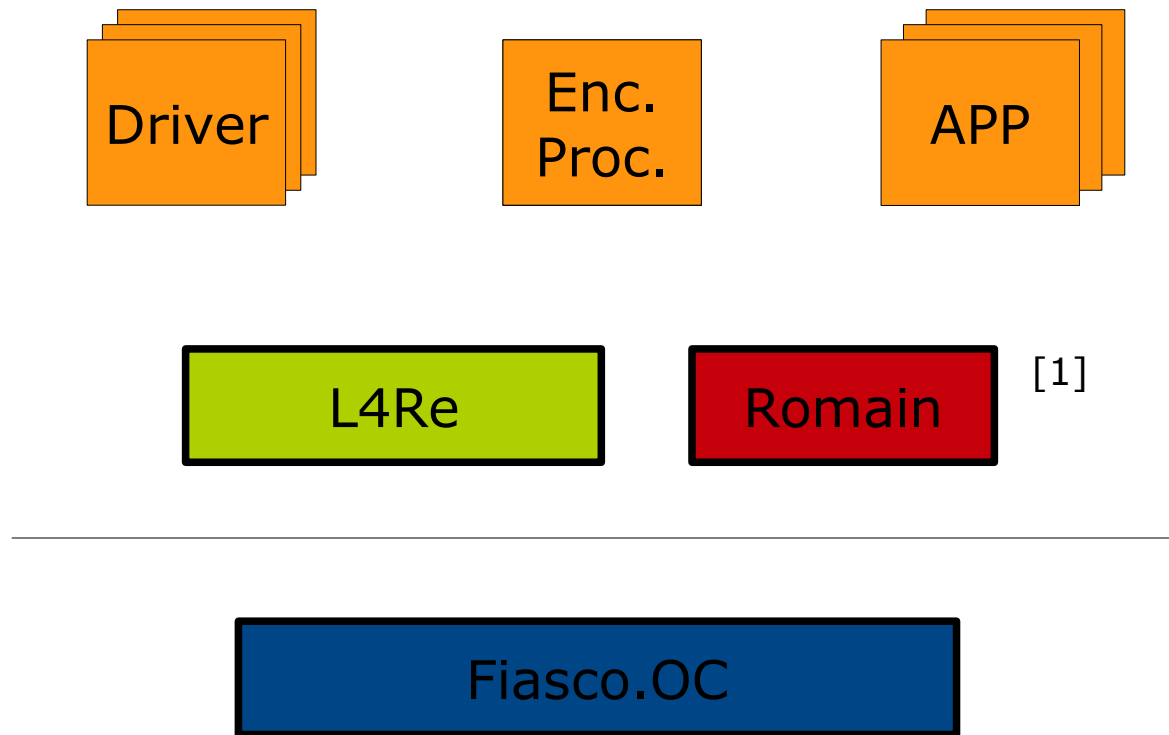
Challenges and Opportunities

[Björn Döbel](#), Robert Muschner, Hermann Härtig
TU Dresden Operating Systems Group

Paderborn, 30.05.2014

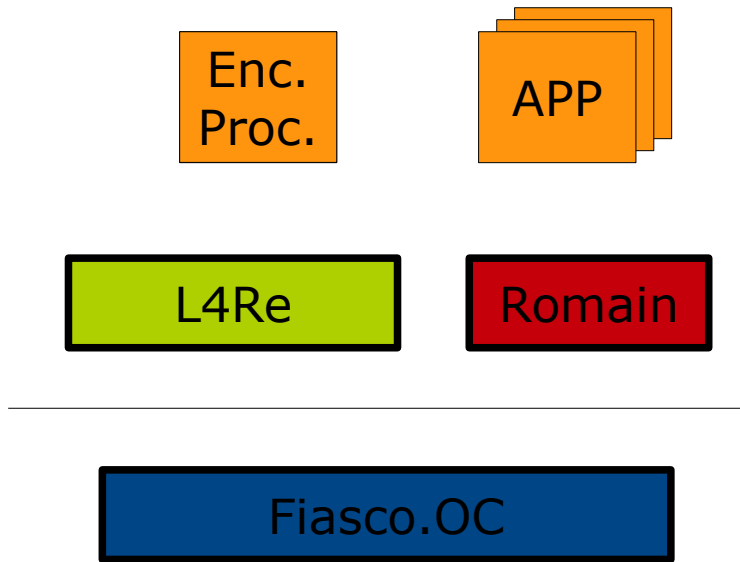


ASTEROID – OS-Assisted Replication



[1] Döbel, Härtig, Engel: *Operating System Support for Redundant Multithreading*, EMSOFT 2012

ASTEROID – OS-Assisted Replication

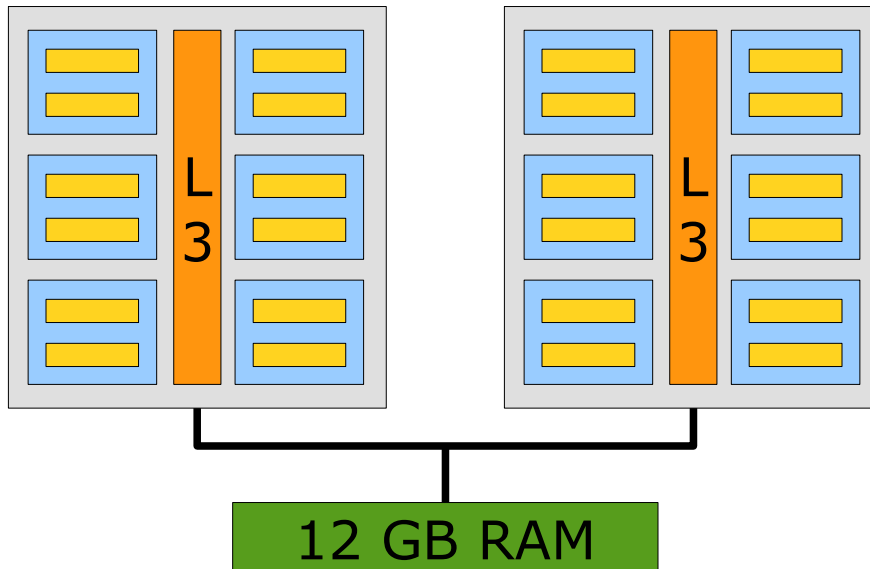


- Interpose on system calls & CPU exceptions
- Replicate memory (no need for ECC)
- Unmodified binary applications

Adapting to Manycore Hardware

- **Adapting to Cache Hierarchy**
 - Goal: decrease runtime overhead
 - This paper: SPEC INT 2006
- **Adapting to Application Vulnerability**
 - Goal: decrease resource consumption
- **Adapting to Hardware Vulnerability**
 - Goal: Protect the Reliable Computing Base

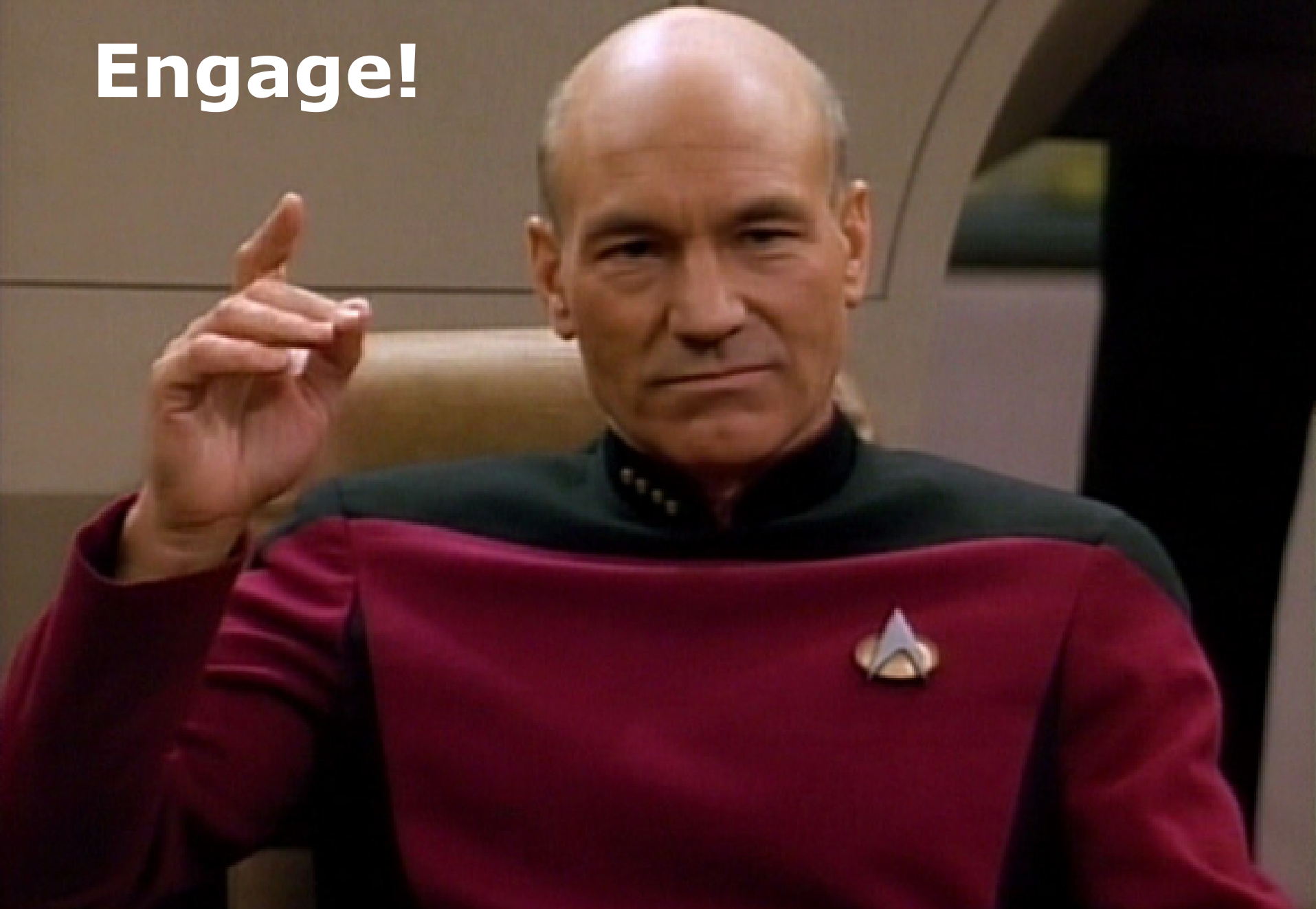
Measuring Runtime Overhead



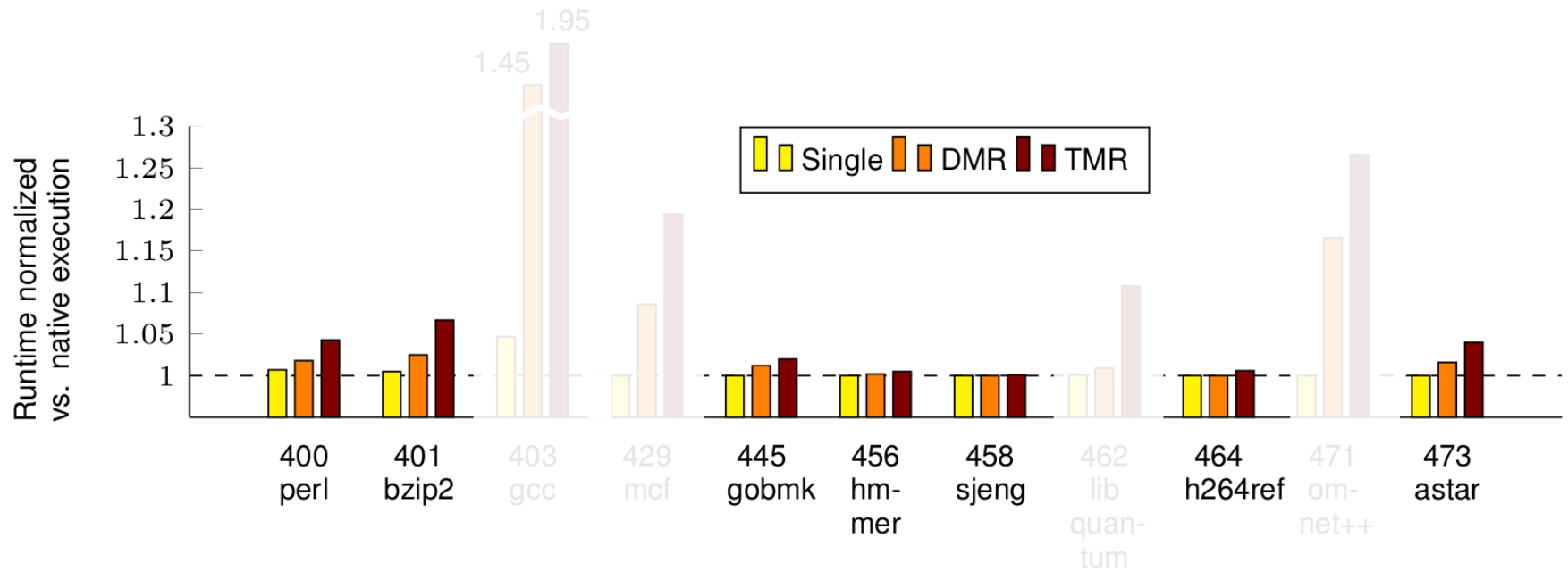
12x Intel X5650
@ 2.66 GHy

- L4/Fiasco.OC,
32 bit + Romain
- SPEC INT 2006
 - 400.perl
 - 401.bzip2
 - 403.gcc
 - 429.mcf
 - 445.gobmk
 - 456.hmmer
 - 458.sjeng
 - 462.libquantum
 - 464.h264
 - 471.omnet++
 - 473.astar
 - ~~478.xalanbmk~~

Engage!



Replicating SPEC INT 2006



Secondary Effects: Cache Miss Rates

Benchmark	DMR misses		TMR misses	
	L2	L3	L2	L3
429.mcf	2,600	1,300,000	11,000,000	5,200,000
462.libquantum	2,500	570	440,000	387,000
471.omnet++	270,000	6,900,000	35,000,000	21,200,000

x 130



Adapting Replica Placement

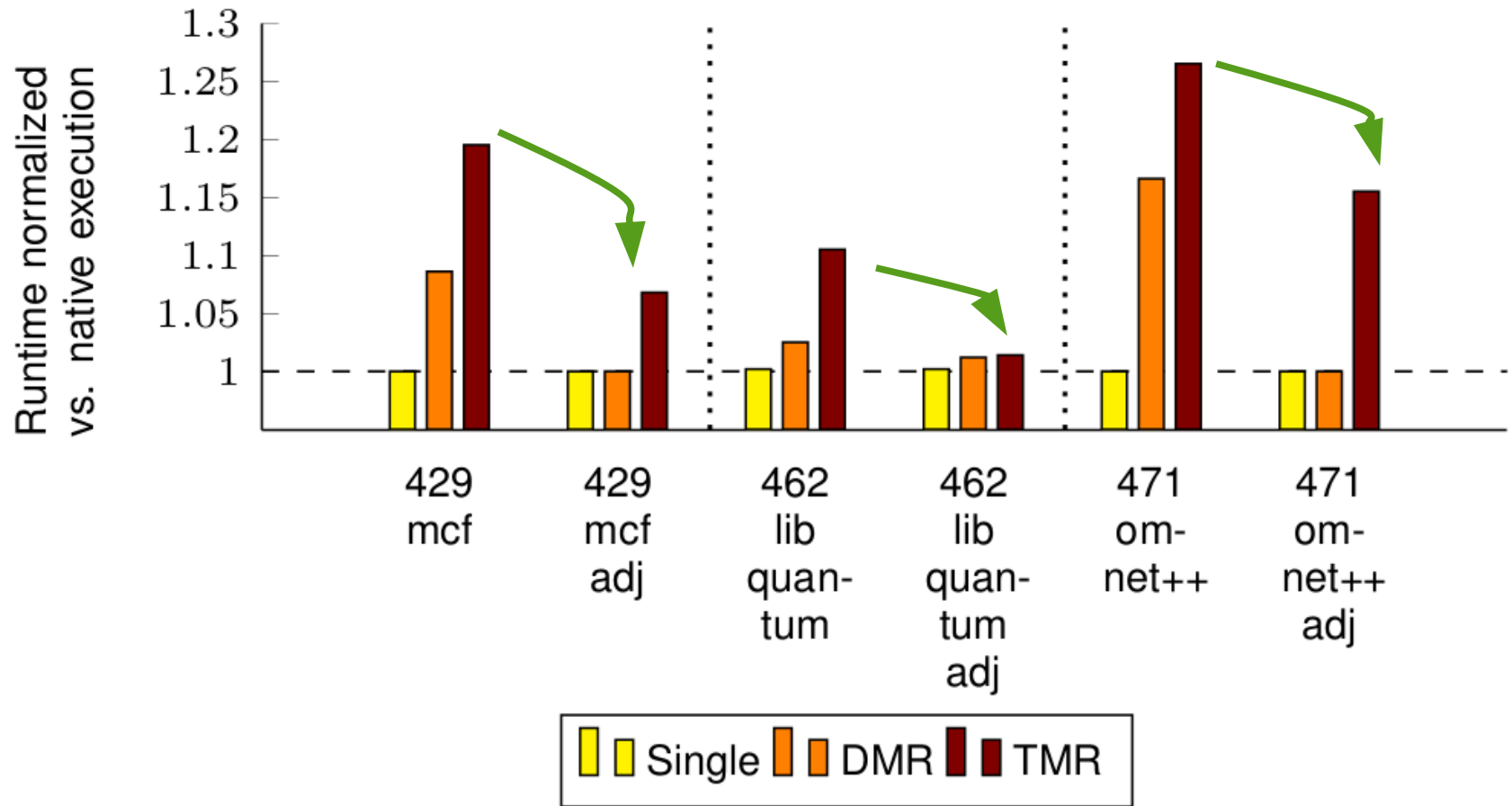
Idea: Run replicas on different sockets

→ improved L3 cache usage

→ may also gain improved fault tolerance

Benchmark	DMR misses		TMR misses	
	L2	L3	L2	L3
429.mcf	2,600 → 2,600	1,300,000 → 930,000	11,000,000 → 11,000,000	5,200,000 → 3,600,000
462.libquantum	2,500 → 2,500	570 → 323	440,000 → 385,000	387,000 → 8,700
471.omnet++	270,000 → 290,000	6,900,000 → 5,500,000	35,000,000 → 34,900,000	21,200,000 → 16,400,000

SPEC INT: Improved L3 Miss Rates



Replication vs. Resource Consumption

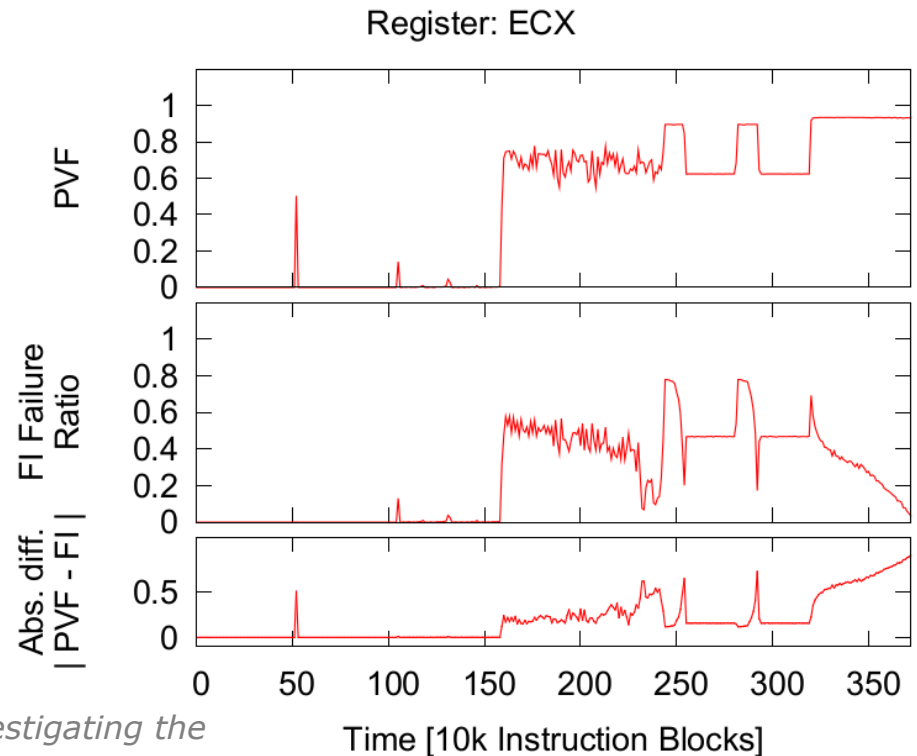
N-Way Modular Redundancy

N CPUs → Nx energy consumption

N x memory usage

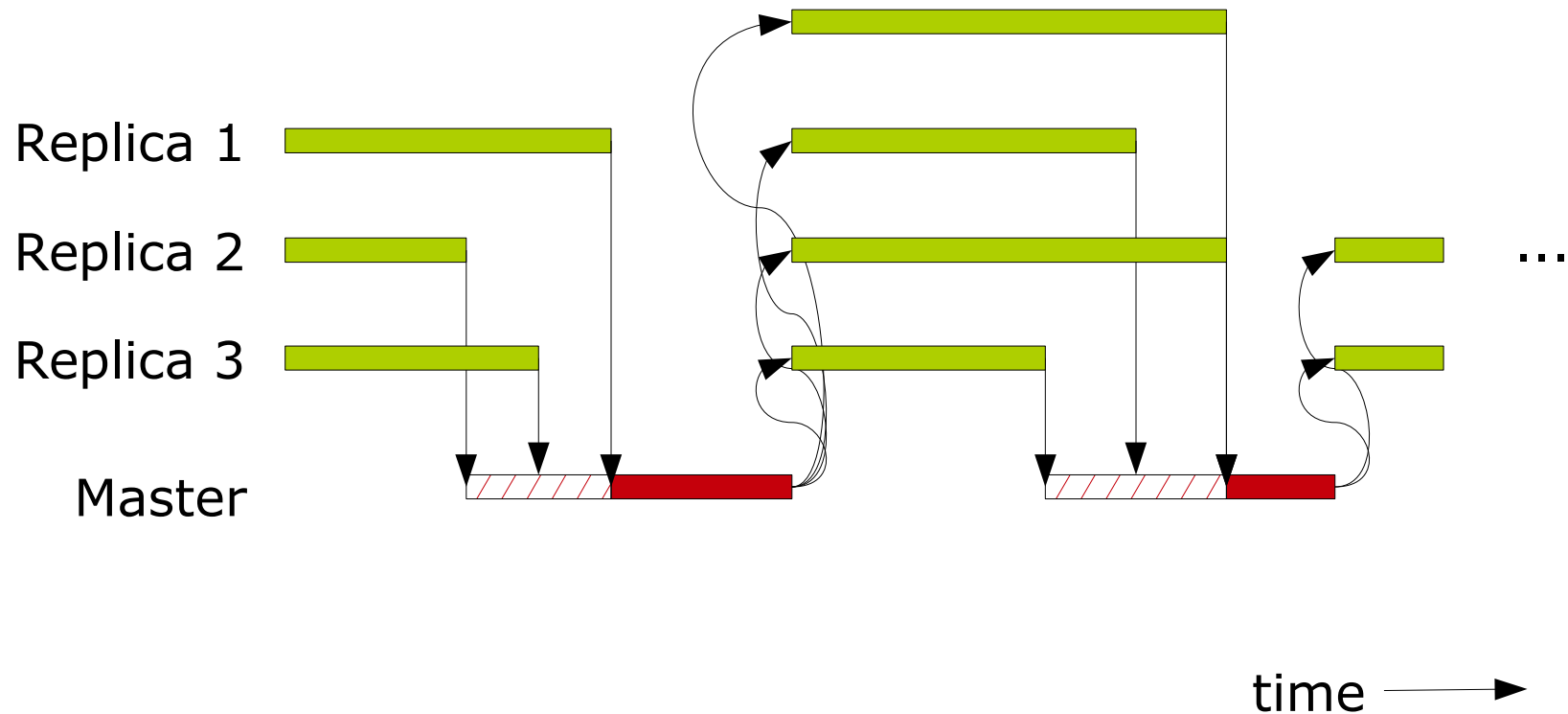
Observation:

Program Vulnerability
is non-constant



[2] Döbel, Schirmeier, Engel: *Investigating the limitations of PVF for realistic Program Vulnerability Assessment*; DFR 2013

Romain: Adaptive Replication



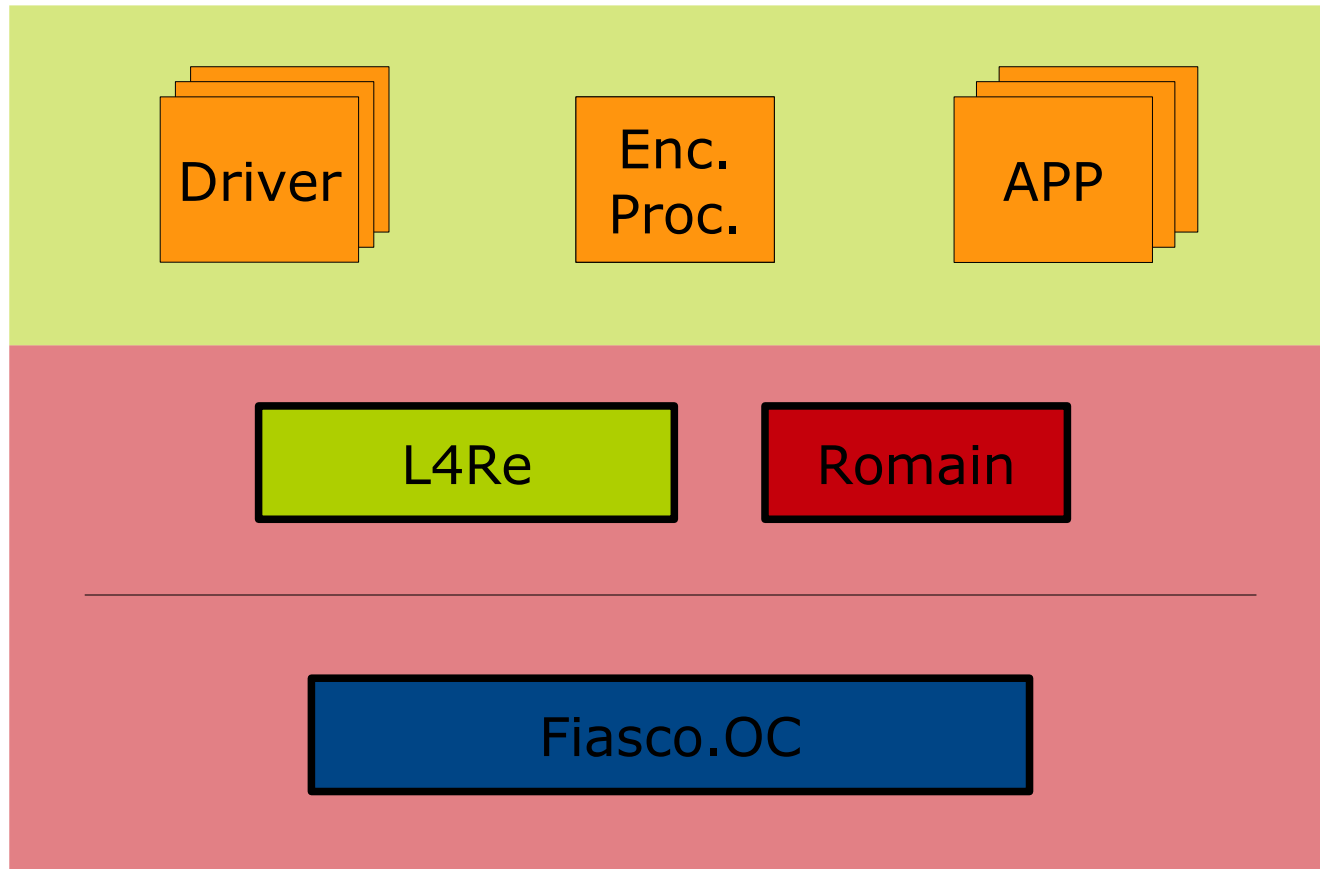
Saving Resources By Adaptation

Fewer replicas → less CPU / energy consumption

Memory?

- Decreasing replica count: need to free resources
 - release in background
- Increasing replica count: need to create new copies
 - copy-on-write memory
 - BUT: must be sure that COW regions are not faulty!

The Reliable Computing Base (RCB)



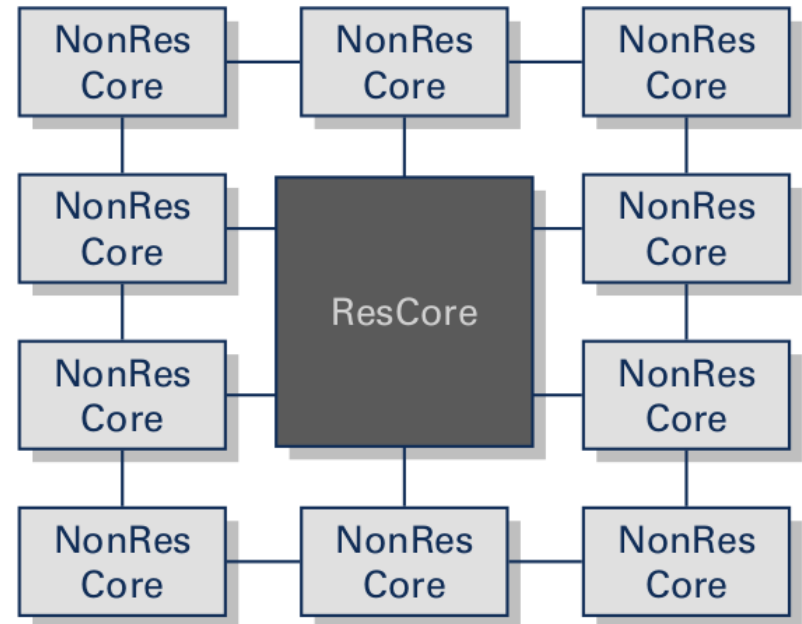
Embracing Hardware Heterogeneity

Heterogeneity today:

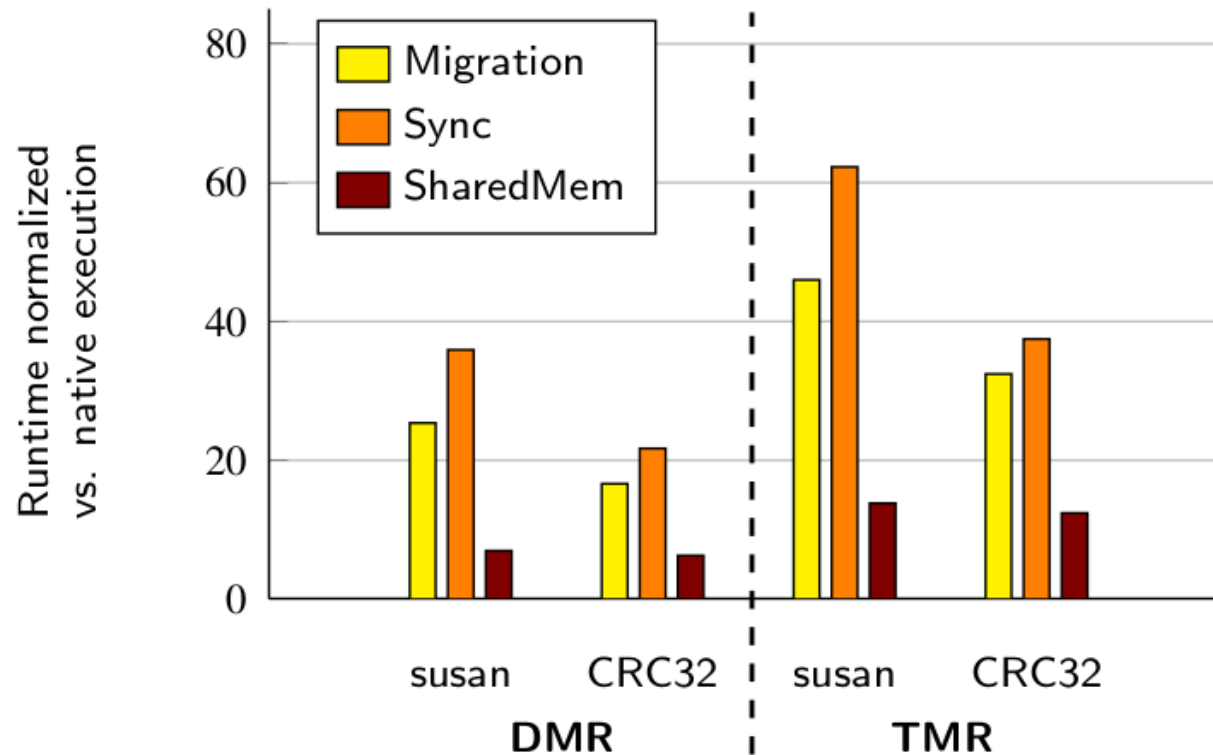
- IBM Cell
- ARM big.LITTLE
- Error-Resilient System Architecture
 - Resilient + non-resilient cores

ASTEROID fits into Res/NonRes view:

- Master runs on ResCores
- Replicas run anywhere



Transitioning Between Res and NonRes Cores



Summary: Adaptation Challenges

- Performance Adaptation
Cache-bound vs. Comm-bound workloads
- Resource Adaptation
Adaptive replication vs. Resource management
- Adaptation to Hardware Vulnerability
RCB protection using resilient cores