



mSWAT: Low-Cost Hardware Fault Detection and Diagnosis for Multicore

Siva Kumar Sastry Hari, Manlap Li, Pradeep Ramachandran, Byn Choi and Sarita Adve

University of Illinois, Urbana-Champaign

Published at 42nd International Symposium on Microarchitecture (MICRO), December 2009



Motivation

Moore's Law \Rightarrow More Transistors with smaller feature sizes \Rightarrow More in-field failures

\Rightarrow **Need in-the-field detection, diagnosis, recovery, repair**

SoftWare Anomaly Treatment (SWAT): comprehensive reliability frame-work

Effective for HW faults in single-threaded apps

But multicore systems w/ multithreaded apps here to stay

Does the SWAT approach work for multicore?

Key Challenge: Cross-Core Fault Propagation

Multithreaded apps share data across threads

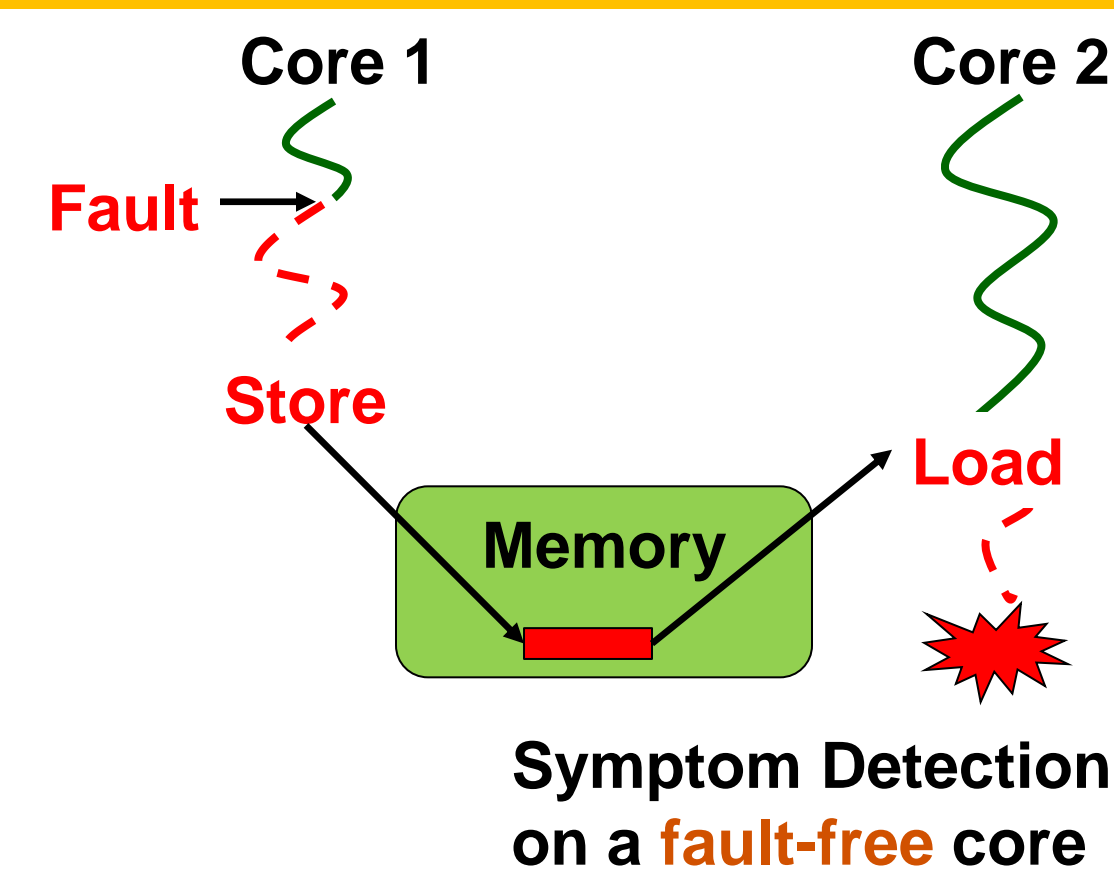
\Rightarrow Fault may propagate across cores

\Rightarrow **Is SWAT effective in detecting these faults?**

\Rightarrow **Symptom causing core is no longer faulty**

Implicit assumption in prior SWAT work

Need to detect fault and diagnose faulty core



mSWAT Fault Detection

Low-Cost symptom detectors:

Fatal Traps, Hangs,
High OS, Kernel Panic

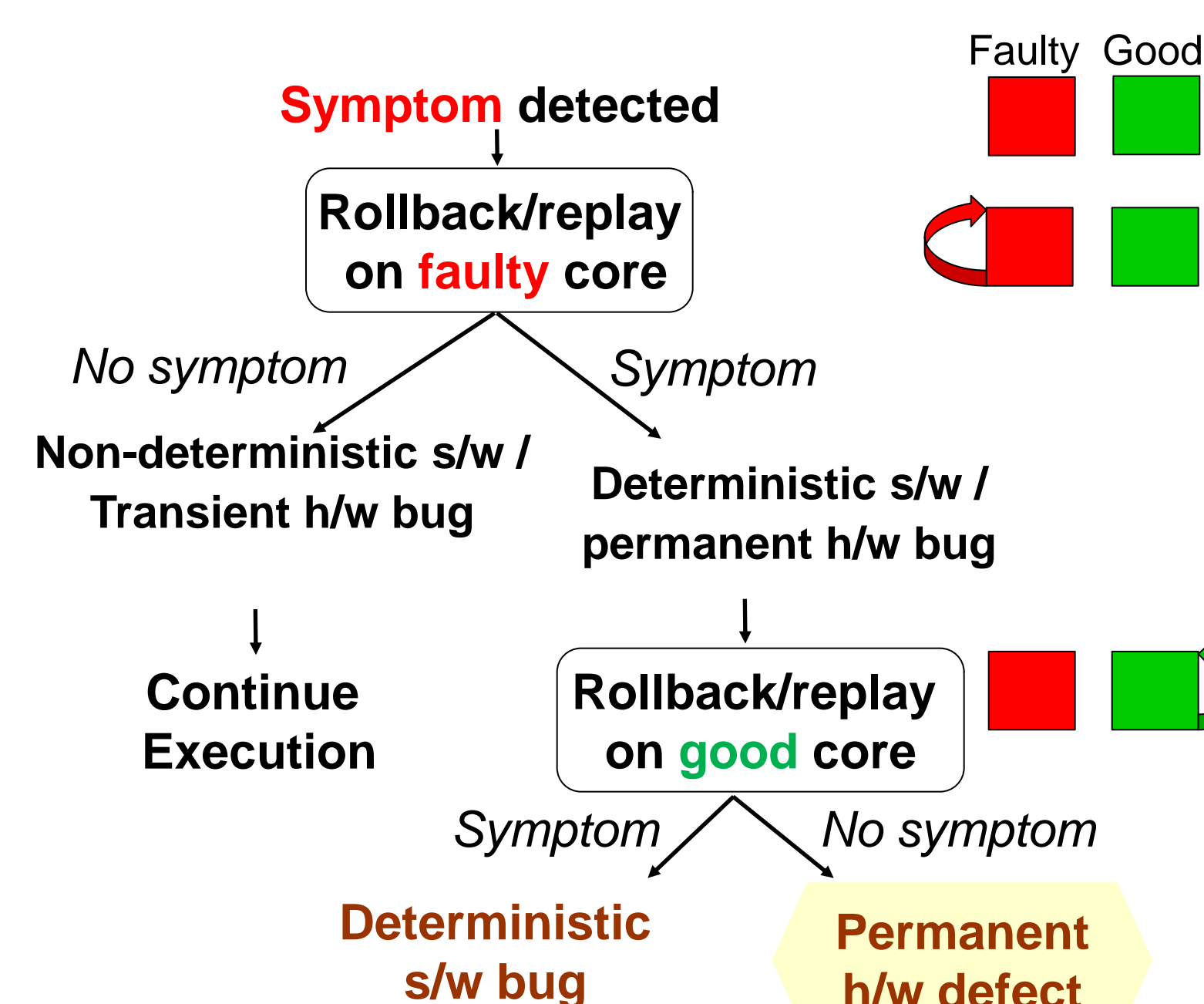
Key Results

Low SDC rate of **0.2%** of injected faults
Several detections from fault-free cores

mSWAT: Diagnosis Challenges and Approaches

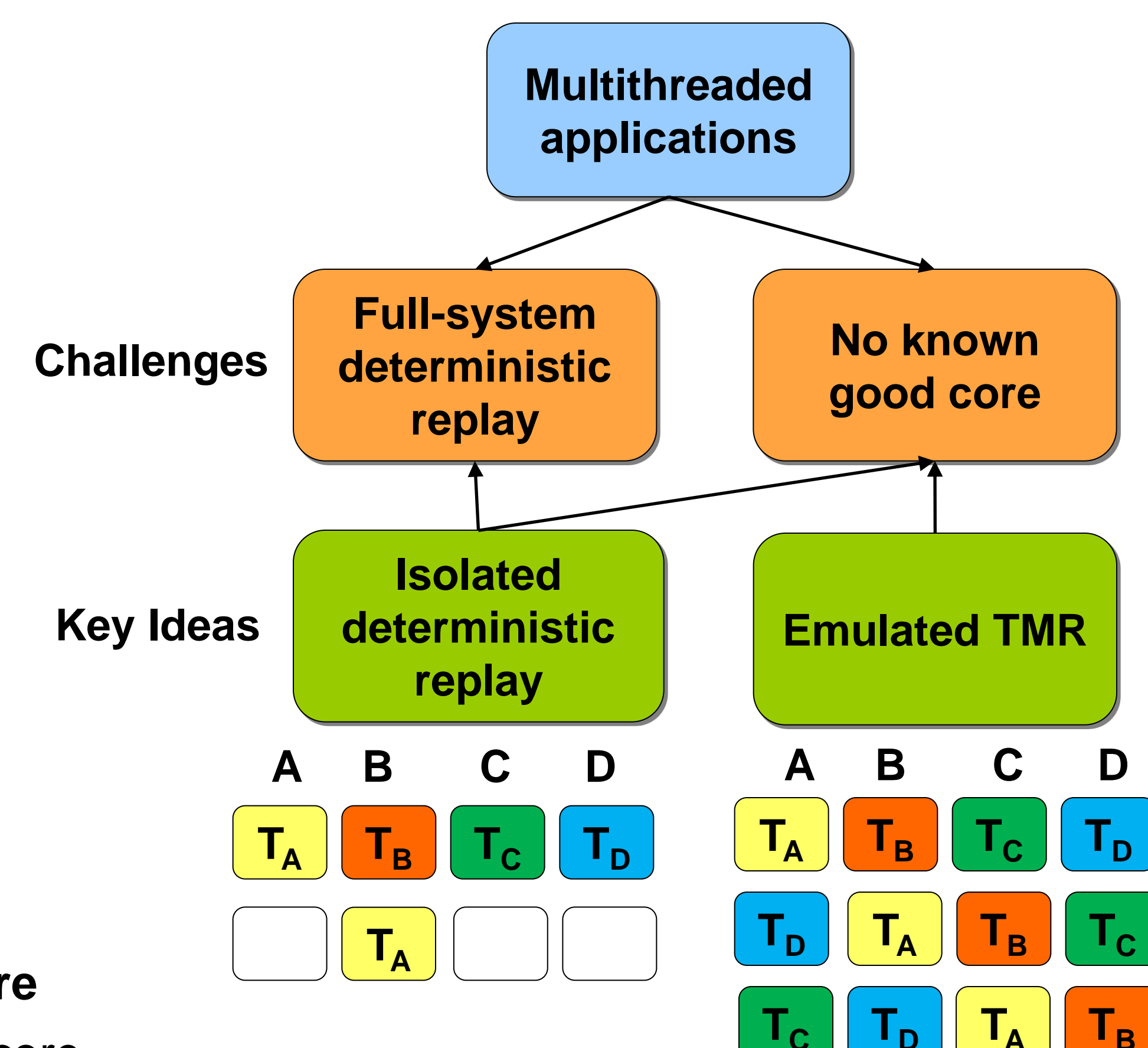
Previous SWAT diagnosis

Distinguish HW/SW faults

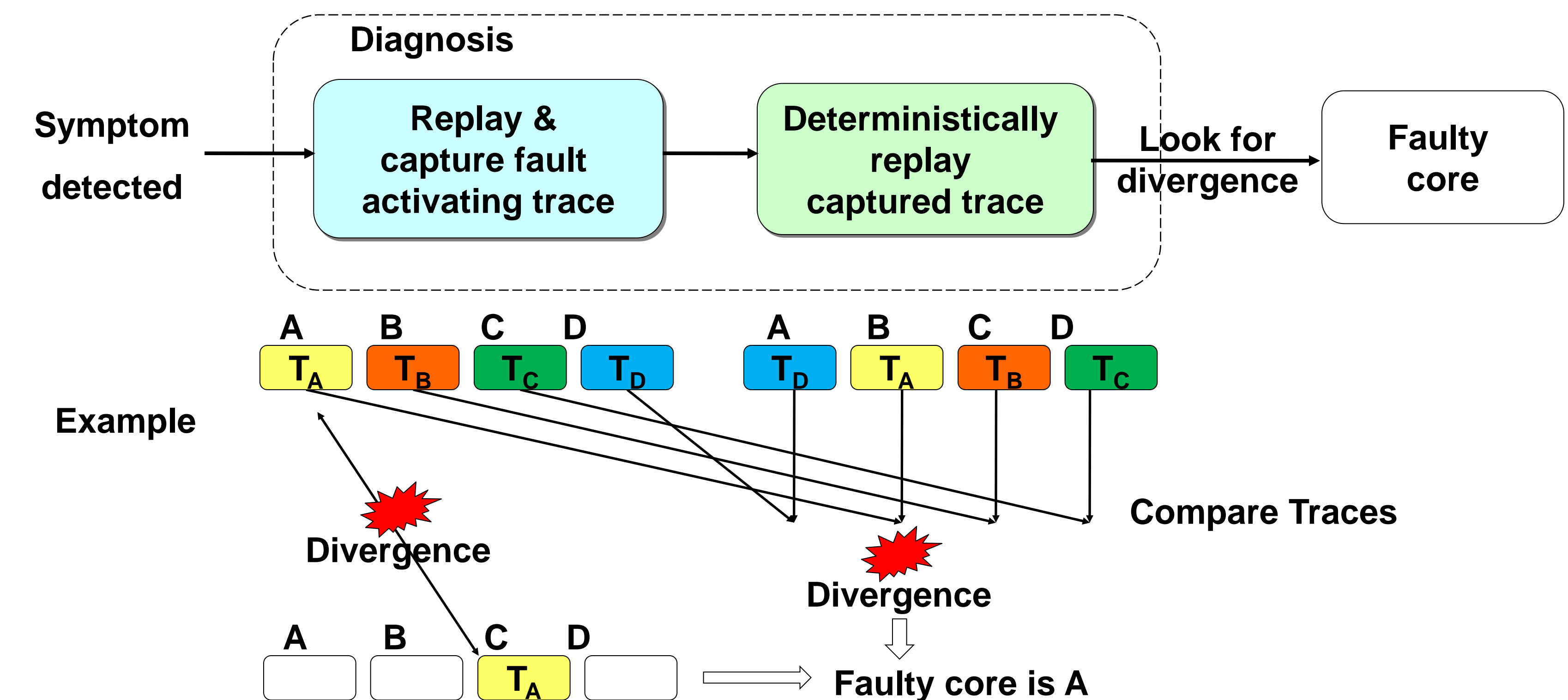


Challenge in multicore: No known good core
mSWAT: Diagnosis without known good core

Isolating the faulty core



mSWAT Fault Diagnosis Algorithm



Capturing fault activating trace

Native execution \Rightarrow No added support for replay

Record inputs to each thread (loads) for replay

Low hardware overhead for buffering

Deterministically replaying captured trace

Firmware emulated isolated deterministic replay \Rightarrow Zero hardware overhead

Compare retiring mem/ctrl instructions for divergence \Rightarrow Fewer comparisons

Iterative Diagnosis to reduce overheads

E.g., capture replay every 100k instructions till divergence

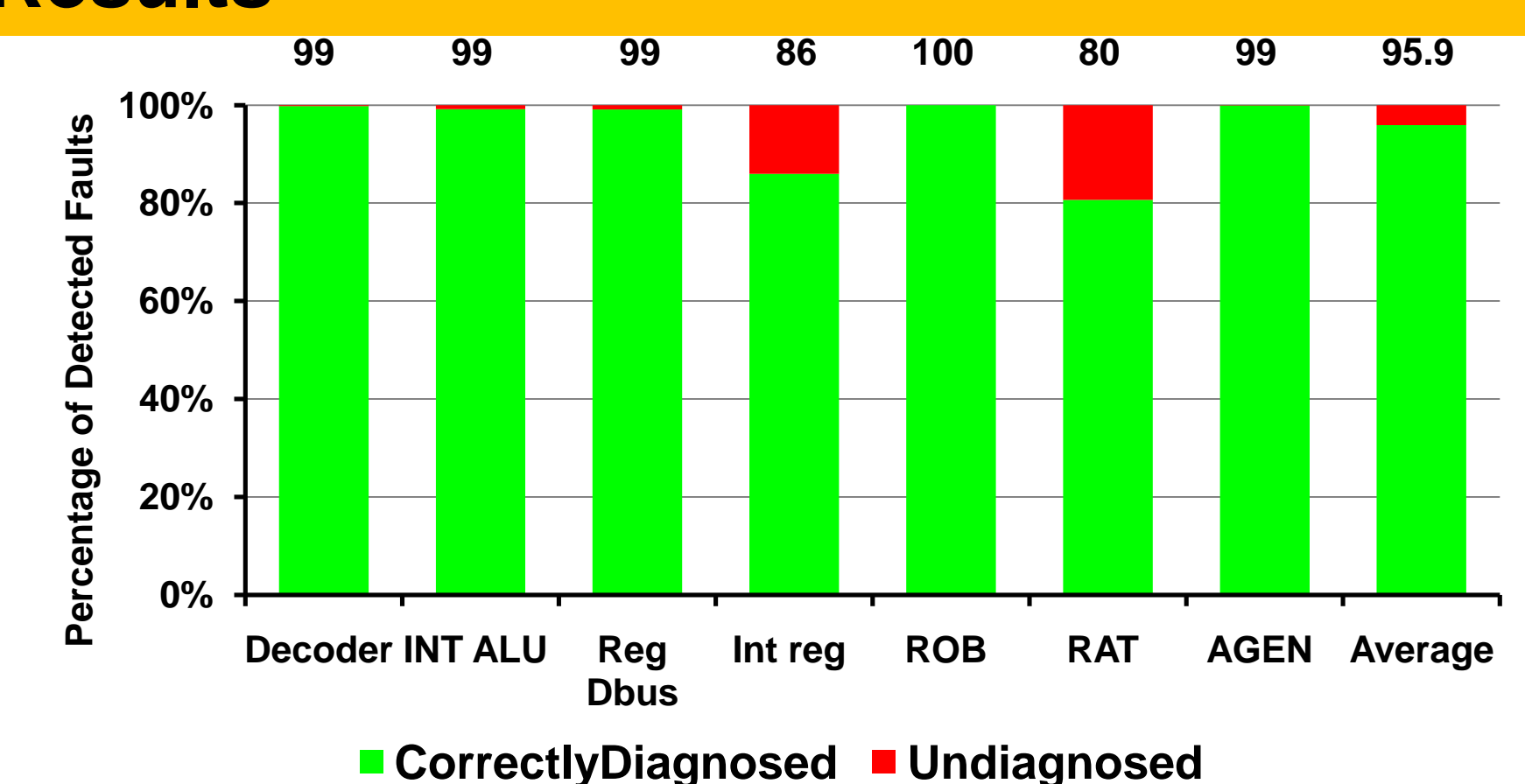
Diagnosis Results

>95% of detected faults successfully diagnosed

97% faults diagnosed in <10M cycles

<10ms on a 1GHz processor \Rightarrow invisible

<200KB logs \Rightarrow fit in lower level caches



Conclusions and Future Work

SWAT detection effective even for multicore systems with multithreaded apps

Novel diagnosis mechanism with minimal hardware changes

Ongoing and Future Work

Prototyping SWAT on FPGA in collaboration with University of Michigan

Faults in off-core components