

The Reliability Threat

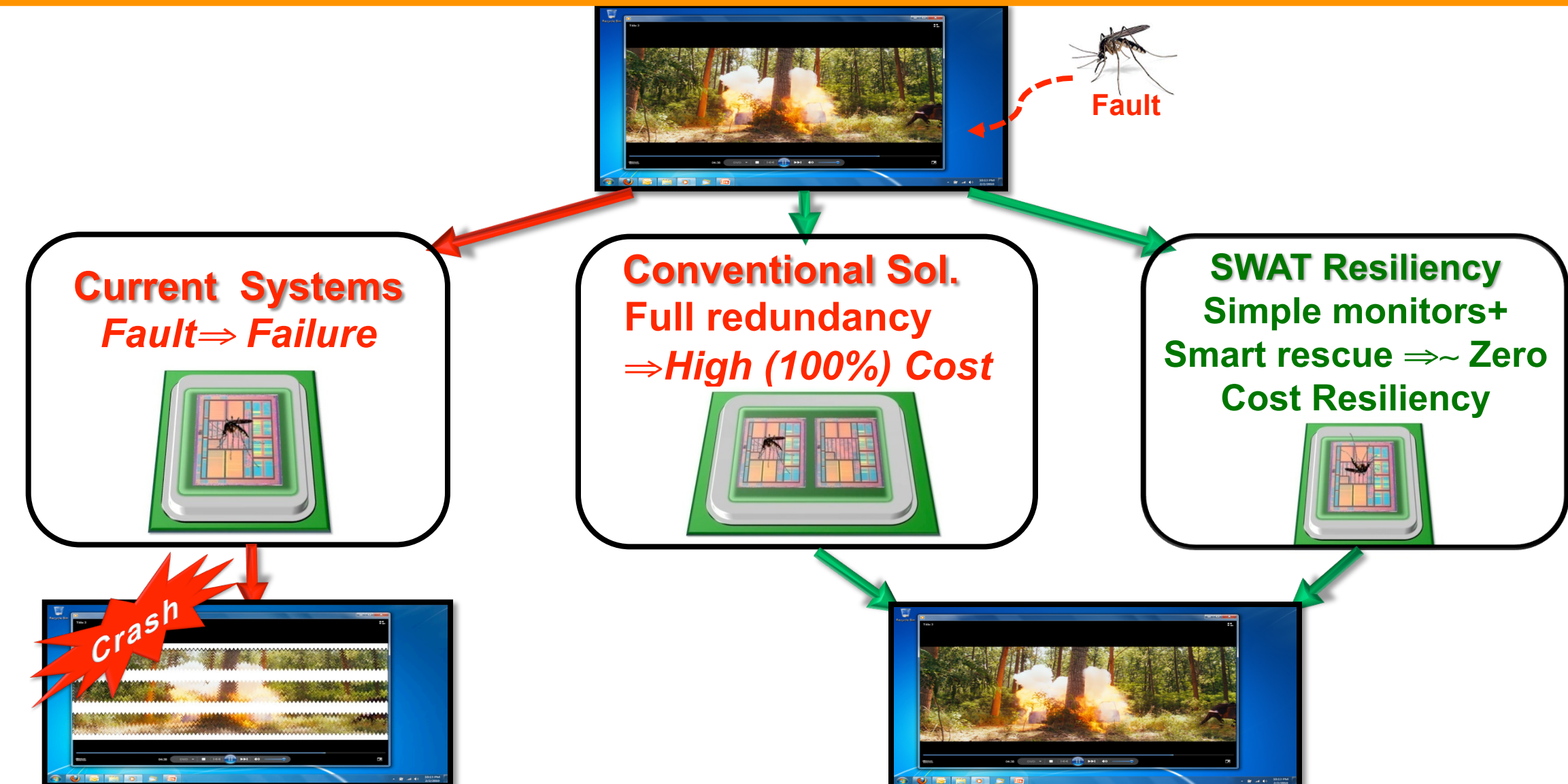
Technology scaling \Rightarrow smaller devices vulnerable to failures
Increased in-the-field failures in commodity systems

Wear-out **Transient errors** **Design Bugs ... and so on**

Need low-cost detection, diagnosis, recovery, repair solutions

Traditional solutions \Rightarrow high area, performance, power

SWAT: A Comprehensive Low Cost Solution



Key Findings

SWAT effective for permanent, transient faults in many apps

Detection: <0.5% SDC rate in SPEC, server, media apps

Low overheads during fault-free execution

Recovery: Majority of faults recoverable in <100K instructions

<5% perf, near-zero area impact from recovery operations

Diagnosis: >95% of detected faults successfully diagnosed

Faulty core identified without spare core

TMR/DMR only for diagnosis \Rightarrow does not impact fault-free exec

Fault Detection [ASPLOS '08, DSN '08]

Goal: Effective, quick detection with minimal fault-free impact

Use symptom detectors to monitor anomalous SW execution

Simple hardware detectors with low area overheads

Fatal Traps

Div by zero, RED state, etc.

Hangs

Simple HW hang detector

Kernel

OS panics due to fault

High OS

High contiguous OS activity

App Abort

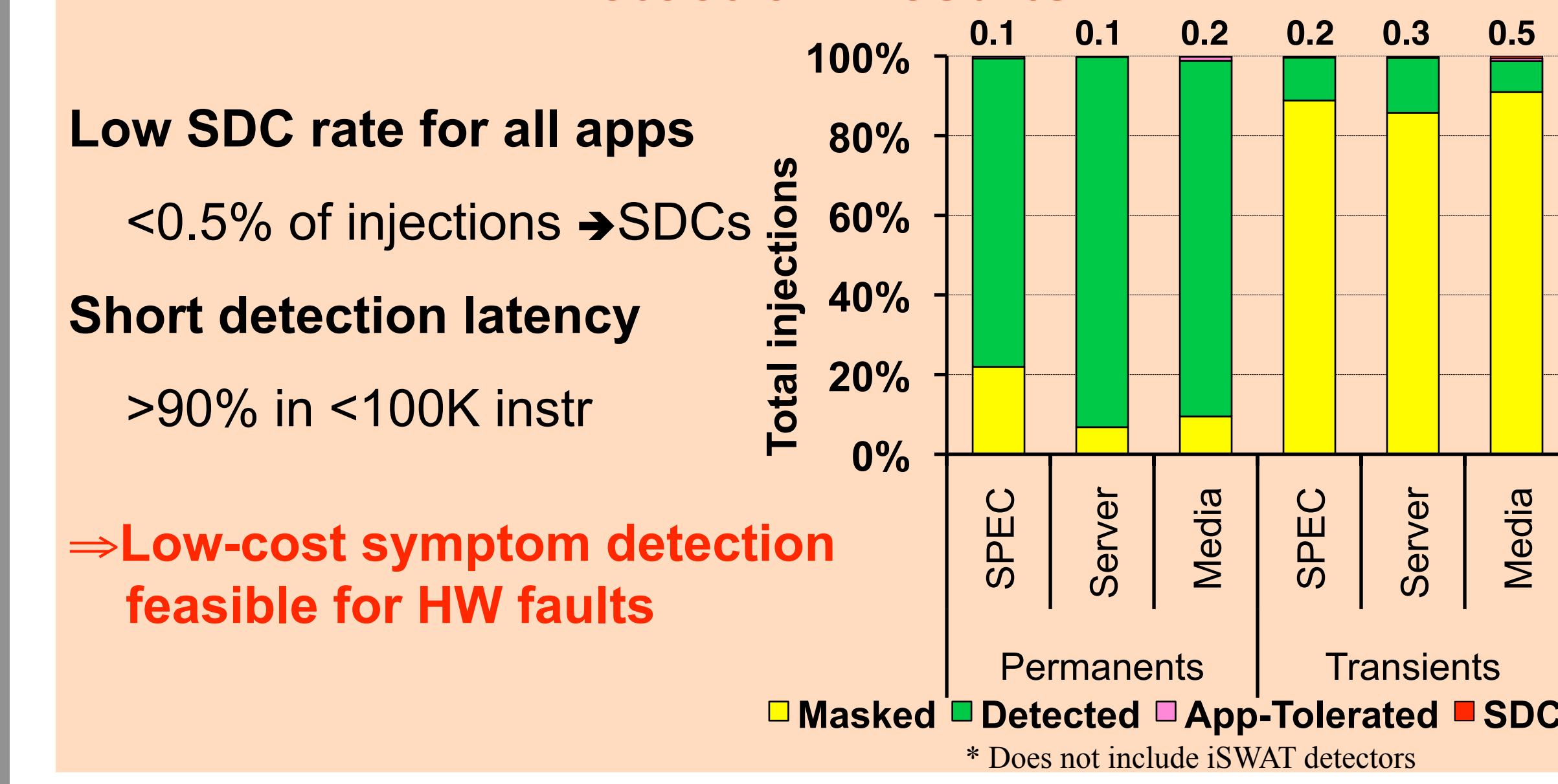
App abort due to fault

Low-cost SW detectors to aid HW detectors

iSWAT
Compiler support to detect faults
Use likely invariants as detectors
Low false +ves, perf. impact

Out-of-Bounds
HW/SW co-designed detector
Monitor legal limit of addresses
Low perf, area overhead

Detection Results



Fault Recovery [submitted]

Goal: Low-cost fault recovery in the presence of I/O

HW checkpoint to restore system state

Low-cost recovery for proc + memory

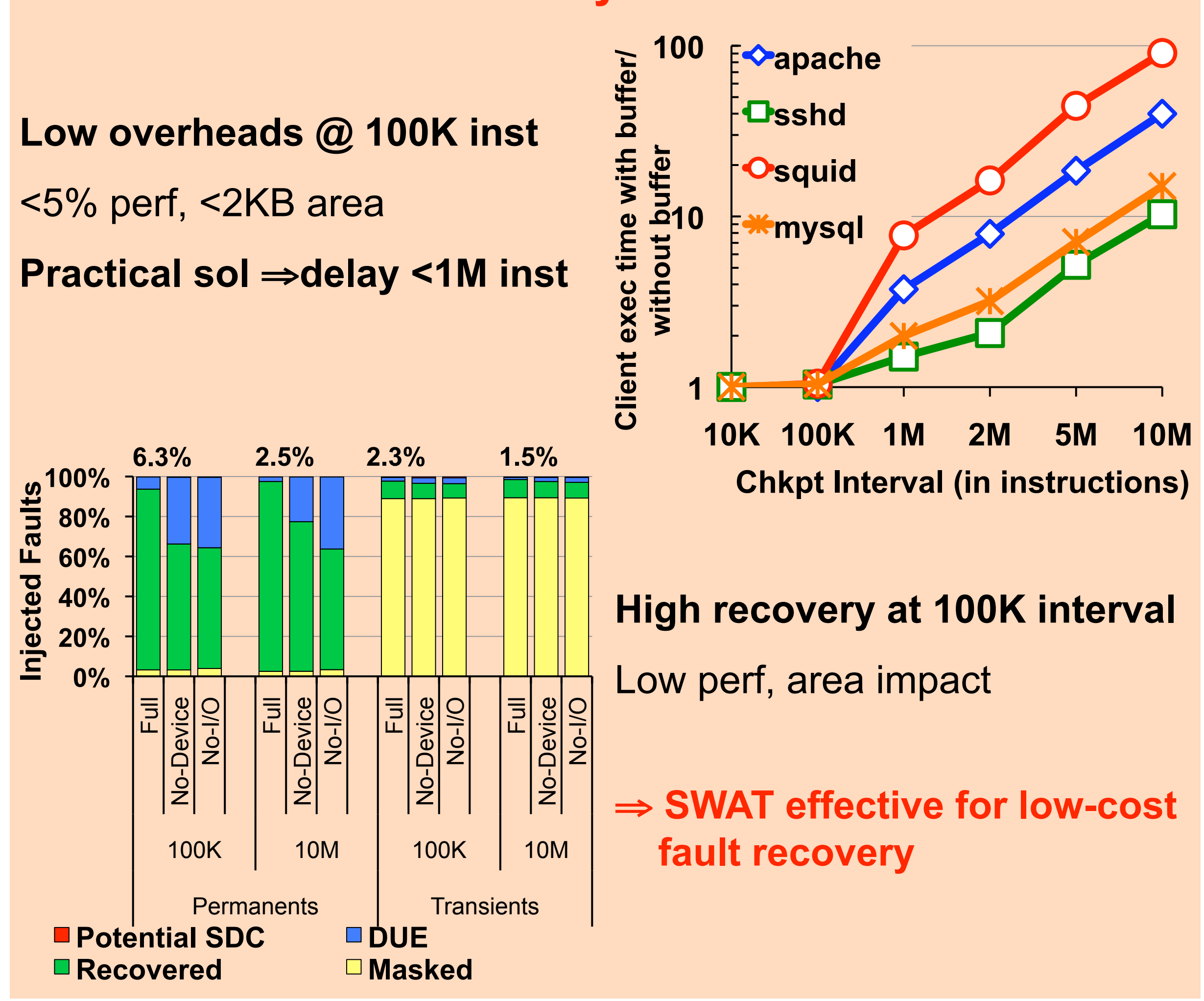
Buffer external outputs in dedicated HW

First low-cost implementation w/ simple HW

Avoids commonly ignored *output-commit* problem

Leverage SW support for device reset, input replay

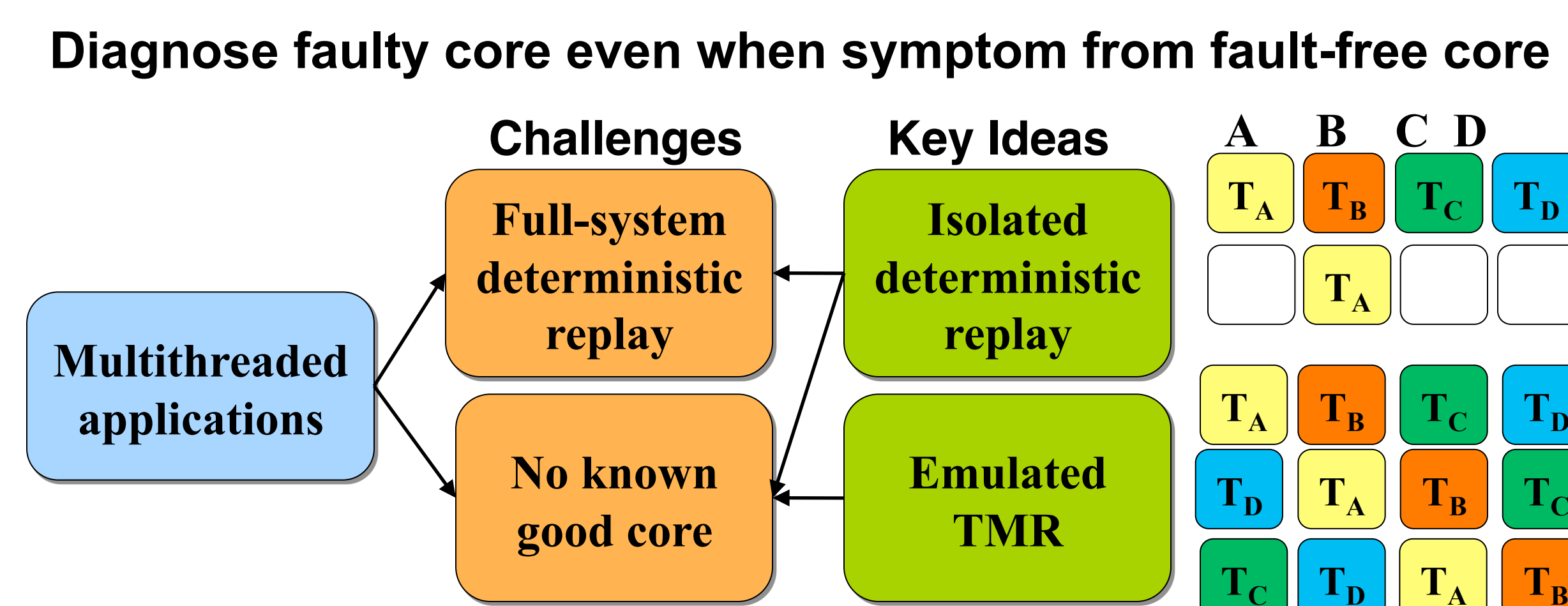
Recovery Results



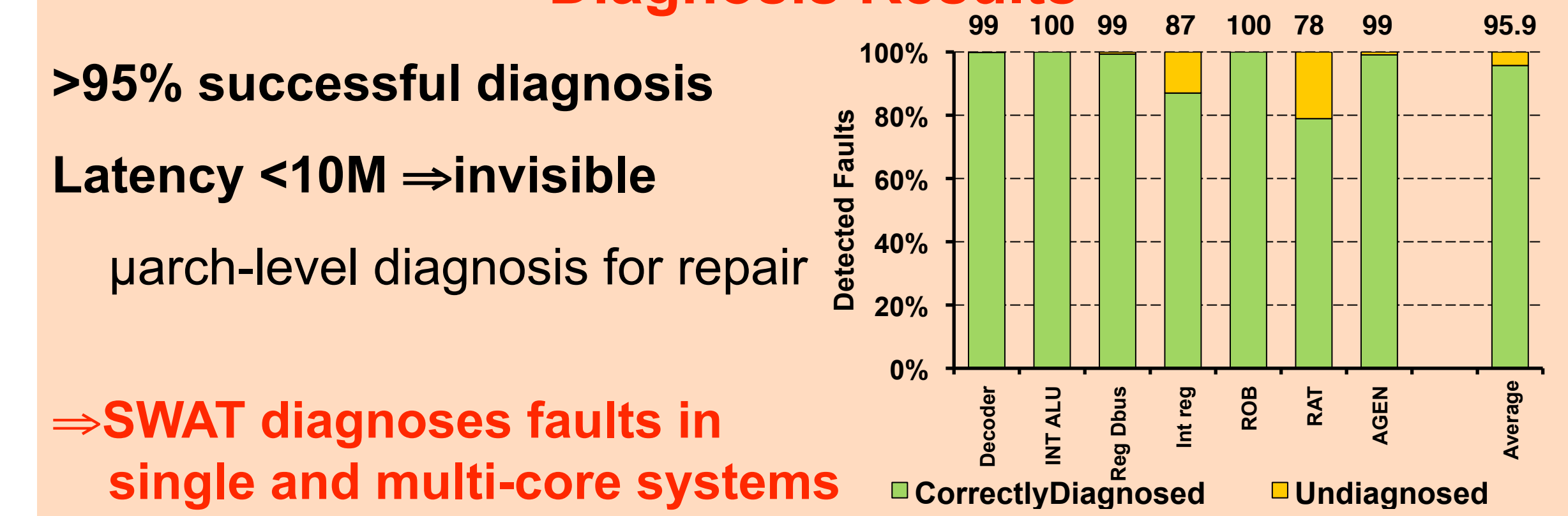
Fault Diagnosis [DSN '08, MICRO '09]

Goal: Diagnose fault source without affecting fault-free exec

\Rightarrow **No spares for diagnosis**



Diagnosis Results



Ongoing and Future Work

Ongoing: Prototyping SWAT on FPGA

Implement SWAT firmware in OpenSolaris

Demonstrate SWAT on multicore OpenSPARC FPGA

Leverage Univ. of Michigan CrashTest for fault injection

Understand when/why SWAT works

Evaluate SWAT for off-core faults, other fault models