BonVoision: Leveraging Spatial Data Smoothness for Recovery from Memory Soft Errors

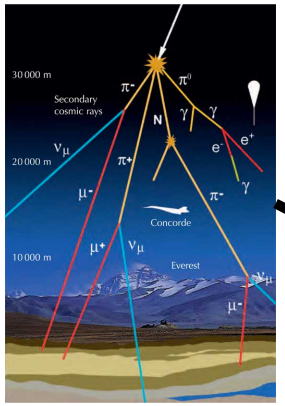
Bo Fang, Hassan Halawa, Karthik Pattabiraman, Matei Ripeanu, and Sriram Krishnamoorthy

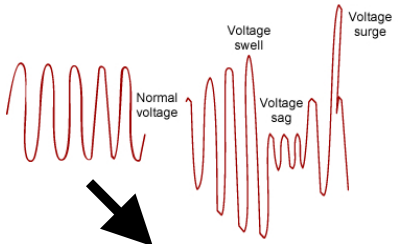
UBC

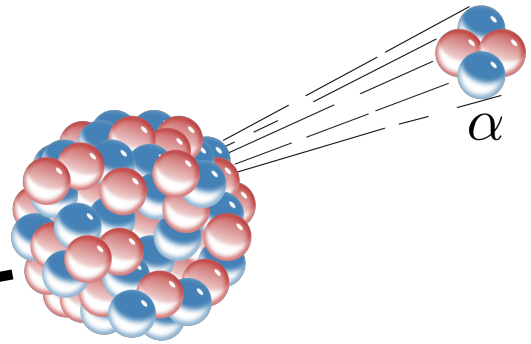
Pacific Northwest NATIONAL LABORATORY

Proudly Operated by Battelle *Since 1965*

Motivation







Transient hardware faults

	Computation	Memory
Fail-stop failures	LetGo [HPDC17']	BonVoision

Memory errors affecting HPC systems

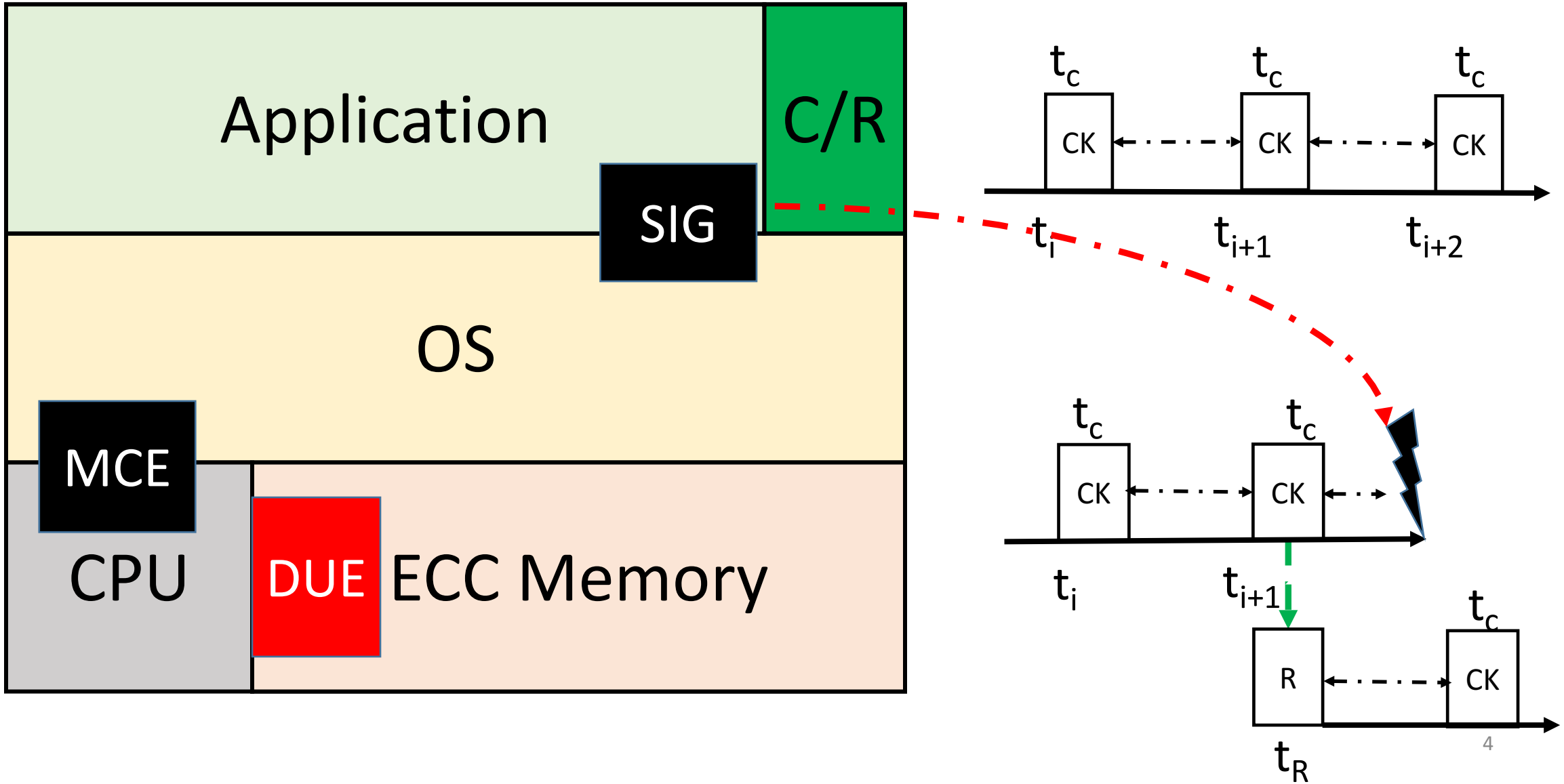
- 260 TB DRAM
- One DRAM error every 11 hour

[Sridharan et al. SC 2013]

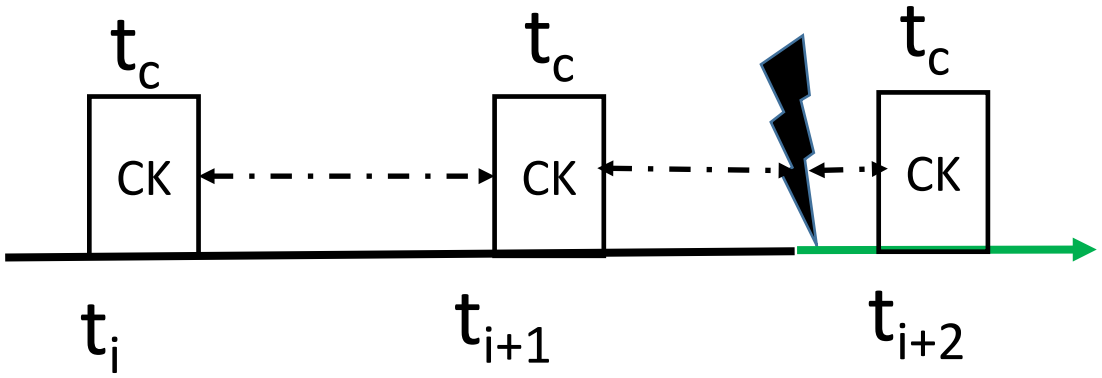
- 1.47 PB RAM
- 160/h memory errors across half of nodes

[Martino et al. DSN 2014]

Handling memory DUEs in software



Roll-forward C/R: mitigate overhead



Benefits:

- longer MTBF, longer ckpt interval
- no restart overhead

Roll-forward recovery for memory DUEs

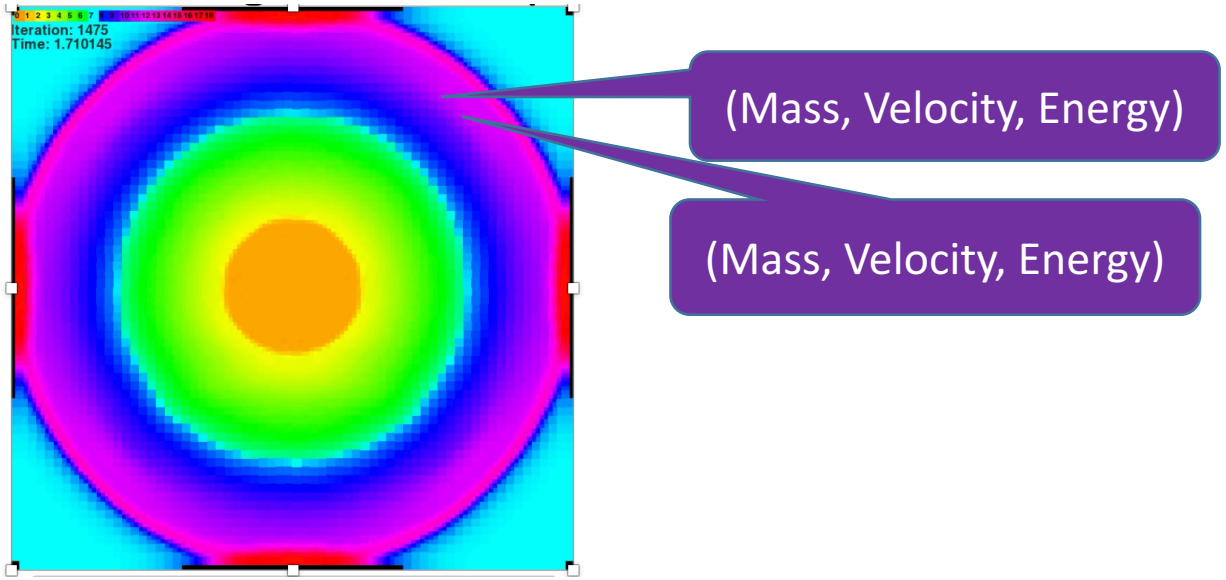
1. **Effectiveness**
2. **Efficiency**

Our approach: BonVoision

Spatial data smoothness

Assumptions:

- Modeled physical space may exhibit inherent continuities



- Logical space mapping to physical space preserves data locality

Smoothness is observed across scientific applications

Exploiting the spatial data smoothness

```
for ( int i = 0; i < n; i ++)
{
    double mass = a[i].m;
  // do some computation
  // ...
}
```

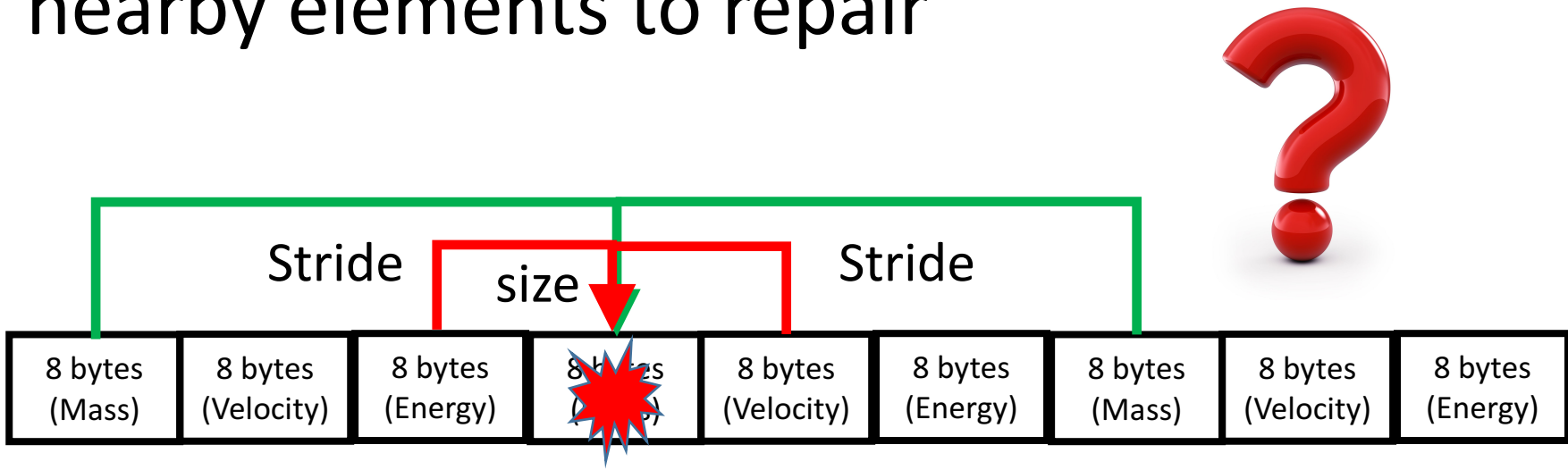
```
4004c8:  addsd  (%rax,%rdx,1),%xmm0
4004cd:  add    $0x10,%rdx
4004d1:  cmp    $0x640,%rdx
4004d8:  jne    4004c8 <main+0x38>
```

DUE

Goal: use values of nearby elements to repair

Available info:

- memory address
- size
- data type



BonVoision: stride speculation

Design space	~~Pr[illegible] an[illegible]n~~	~~[illegible]ne [illegible]g~~
Our approach	<u>Stride speculation</u> Binary analysis Triggered only by DUE	

BonVoision: stride speculation

Key insight

- Different elements are accessed with the same addressing mode
- Modification on the some registers infer

Stride register

```
4004c8:  addsd  (%rax,%rdx,1),%xmm0
4004cd:  add    $0x10,%rdx
4004d1:  cmp    $0x640,%rdx
4004d8:  jne    8  <main+0x38>
```

Stride

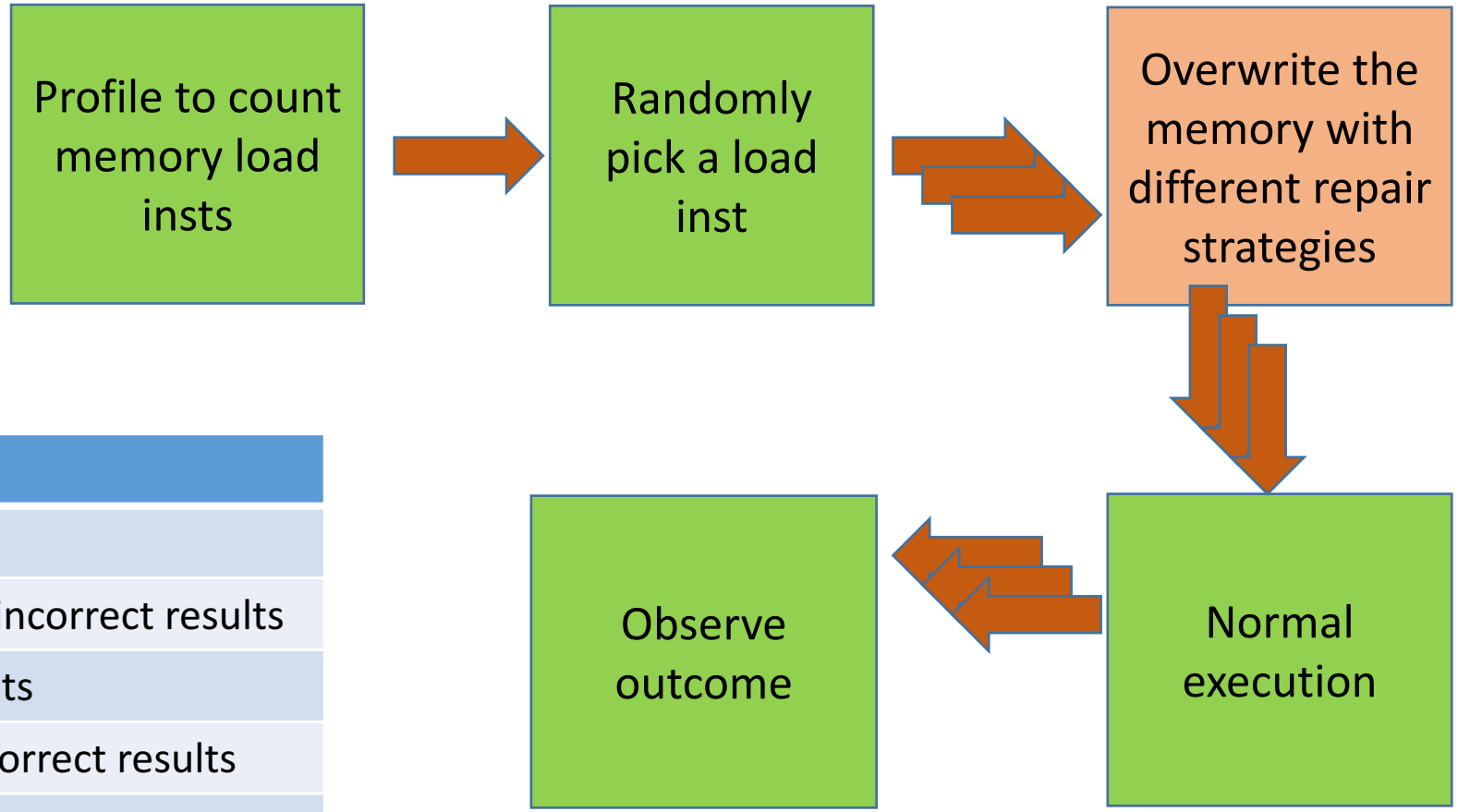
Evaluation questions

- Q1: How effective is BonVoision to recover applications from memory DUEs?
  - Can machine learning help improve BonVoision's effectiveness?
- Q2: How efficient is BonVoision?
- Q3: What is the impact of BonVoision on the C/R?

Evaluation methodology

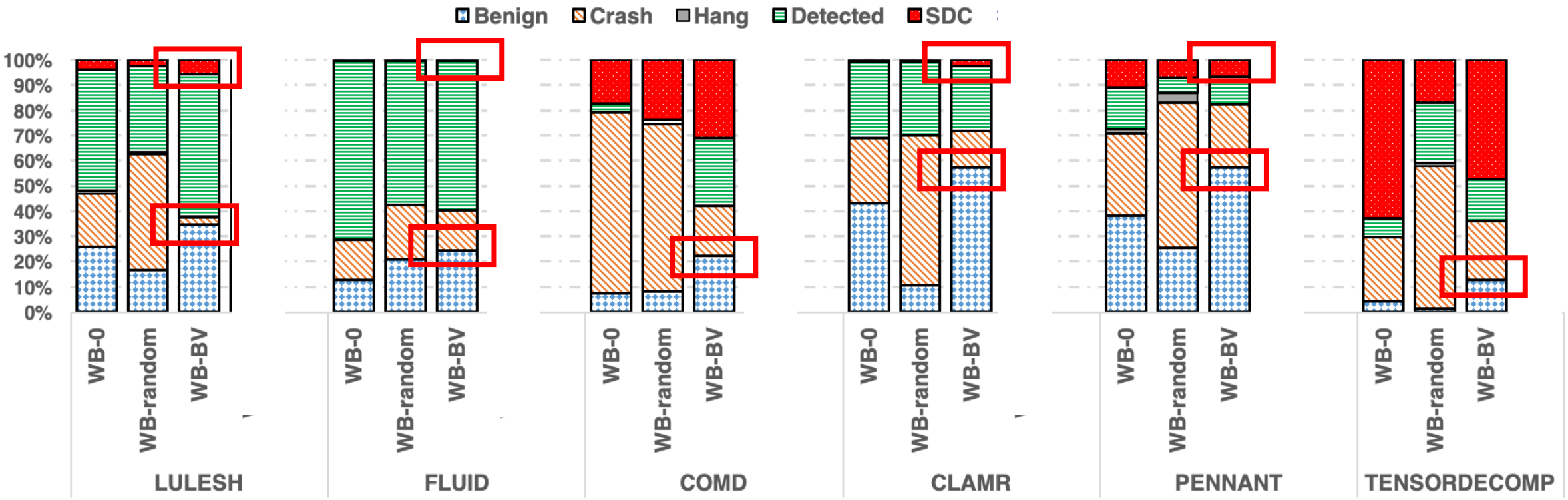
- ***Fault injection: Q1,*** **Q2**
- Simulations: Q3

Fault injection methodology



Outcome	Description
Crash	Exception
SDC	Undetected incorrect results
Benign	Correct results
Detected	Detected incorrect results
Hang	Time out

Evaluation results



BonVoision:

- highest rate of benign outcomes
- 0.1% to 2% increase in SDC rates for four benchmarks

BonVoision-E: optimization with ML

Problem: SDC and detected outcomes

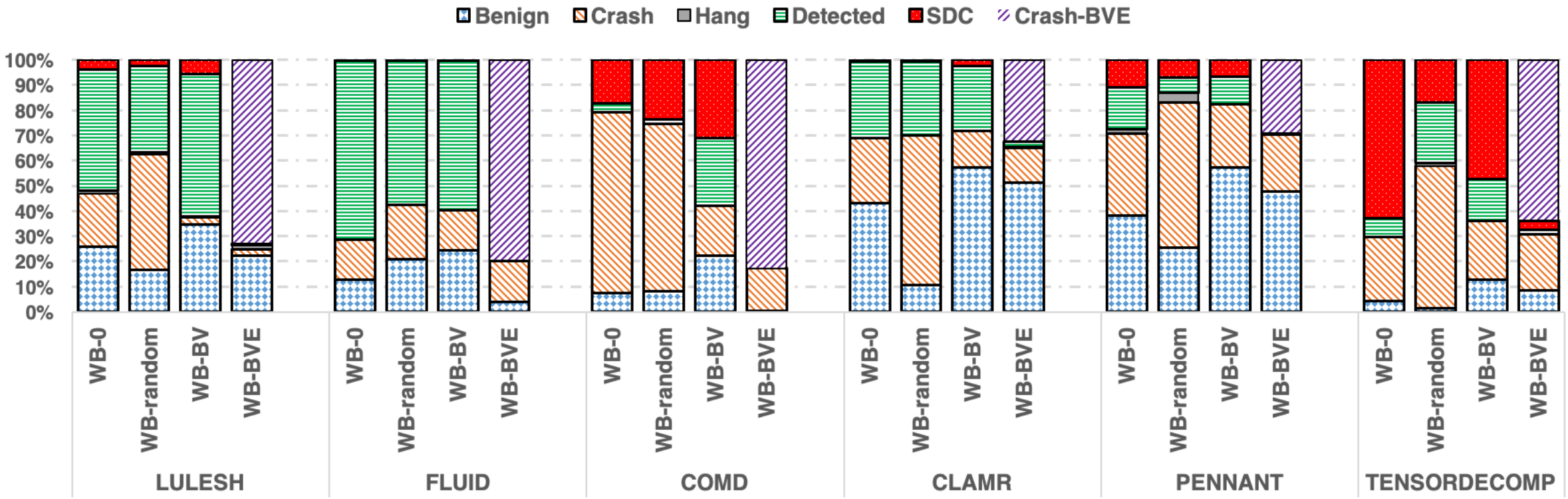
- Detected cases cause applications to waste work

Goal: predict the outcome of a repair online

- If SDC or detected, stop
- If benign, continue

Methodology: supervised machine learning

Online Classifier Results



BonVoision-E converts on average 97% of not benign cases to crashes

Summary

Spatial data smoothness for recovery memory DUEs

A light-weight, automatic, and effective optimization for C/R

Trade off confidence in results for efficiency

Bo Fang's website: https://flyree.github.io/
Netsyslab: http://netsyslab.ece.ubc.ca
DependableSystemsLab: https://github.com/DependableSystemsLab