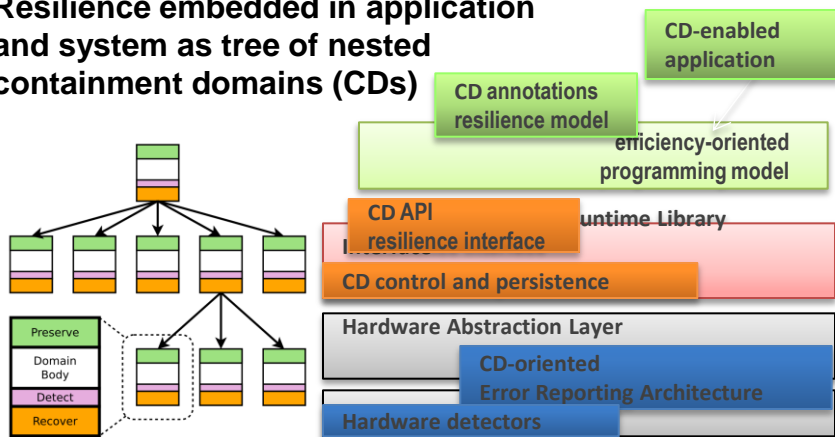


Containment Domains: Programming and Execution Model Support for Resilience

Resilience embedded in application and system as tree of nested containment domains (CDs)



Novel Ideas

- Express resilience as tree of containment domains
 - Each CD localizes resilience actions
 - Distributed and uncoordinated
 - Utilizes entire machine hierarchy
- Incorporate both system and algorithmic detection into abstraction and semantics
 - Cost-efficient elimination of damaging silent data corruption; enable and encourage algorithmic resilience
- Auto-tuned and customized recovery
 - Machine/application cooperative resilience

Impact and Champions

- **Problem:** resilience schemes inefficient or ineffective at extreme scale
- **CDs:** Hierarchical, distributed, and proportional preservation and recovery scales efficiently
- **Problem:** high likelihood of silent data corruption
- **CDs:** Integrated, customizable, and tunable detection; cooperative between algorithm and system
- **Problem:** resilience is ad-hoc and difficult to analyze
- **CDs:** Concise abstraction in application and throughout the system; analyzable and auto-tunable

Principal Investigator(s): Mattan Erez, UT Austin

Milestones/Dates/Status

	<u>Scheduled</u>	<u>Actual</u>
• Node recovery plan	JAN 2013	JAN 13
• Detection abstractions	JUL 2013	JUL 13
• Mini-app evaluation	JUL 2013	JUL 13
• Analytic model	JAN 2014	
• FG kernel-level CDs	JUL 2014	
• Reporting abstractions	JUL 2014	
• Recovery abstractions	JAN 2015	
• Language support	JUL 2015	
• Analysis + Optimizations	JAN 2016	
• Tasking models	JUL 2016	
• Evaluation	JUL 2017	

