www.bsc.es

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Build it for Fault Tolerance, Get Security for Free

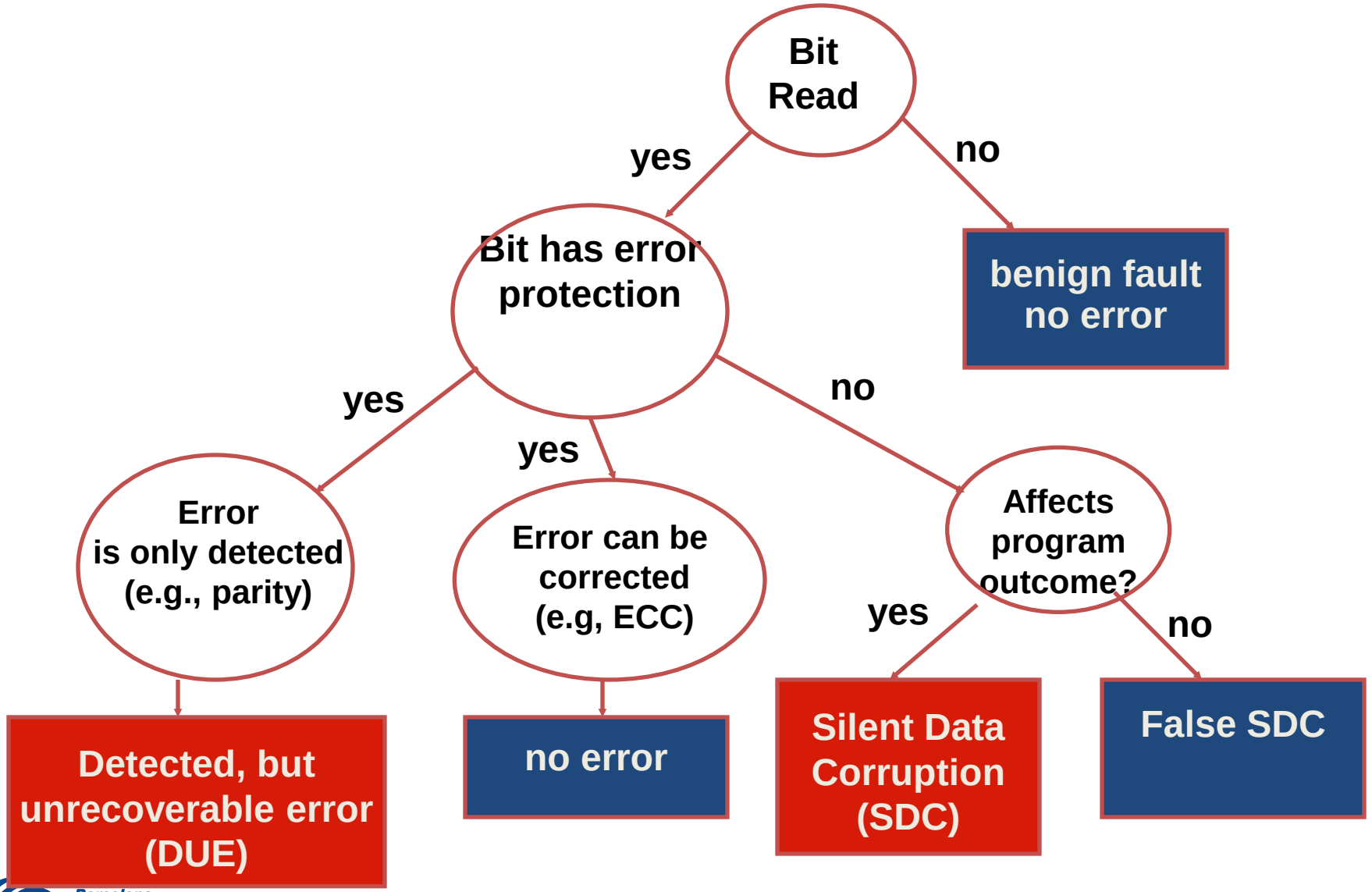
Osman Sabri Ünsal
BSC

HESES Workshop, HIPEAC
January 18th 2016

Fault-tolerance Quick Glance

- Transient Errors
  - Soft errors due to particle strikes
- Intermittent Errors
  - Thermal stress related
- Permanent Errors
  - Burnin
  - NBTI
  - Electromigration
  - TDDB

Transient Error Categorization



Fault-tolerance mechanisms

- Replication of Data
- Replication of Execution (in time, or in space)
  - Duplex
  - Triplex – Majority Voting
- Checkpointing
  - System/application, coordinated/asynchronous, monolithic/incremental, single/multi level
- Coding – Parity, ECC, CRC, RAID, Residue

Fault-tolerance mechanisms

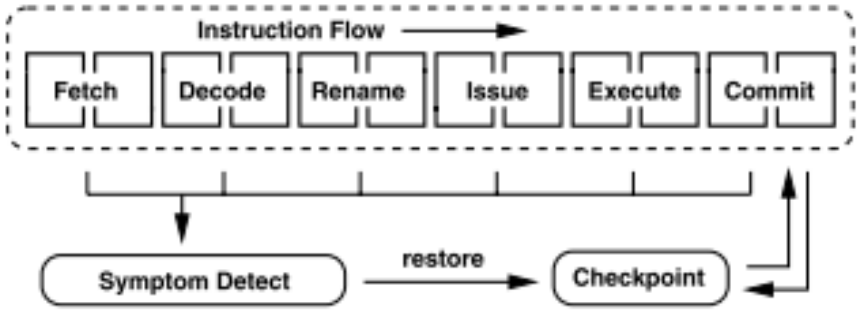
- Error detection
- Error correction
- Fault prediction
- Fault containment

Fault Tolerance → Security

- Fault tolerance techniques typically employ
  - Runtime Checks
  - Ensure correct control of flow
- These are naturally good for security mechanisms, protocols as well
- The relationship is not always mutually beneficiary
- Several case studies follow

1) Symptom-based fault detection

- Transient Faults can be detected by their **symptoms** before they manifest themselves as errors
  - Clustered Last Level Cache misses
  - Excessive branch mispredictions
  - Control flow violations
  - Exceptions



- Wang&Patel, Restore: Symptom Based Soft Error Detection in Microprocessors, DSN 2005
- Intel instruction replay technology in Itanium

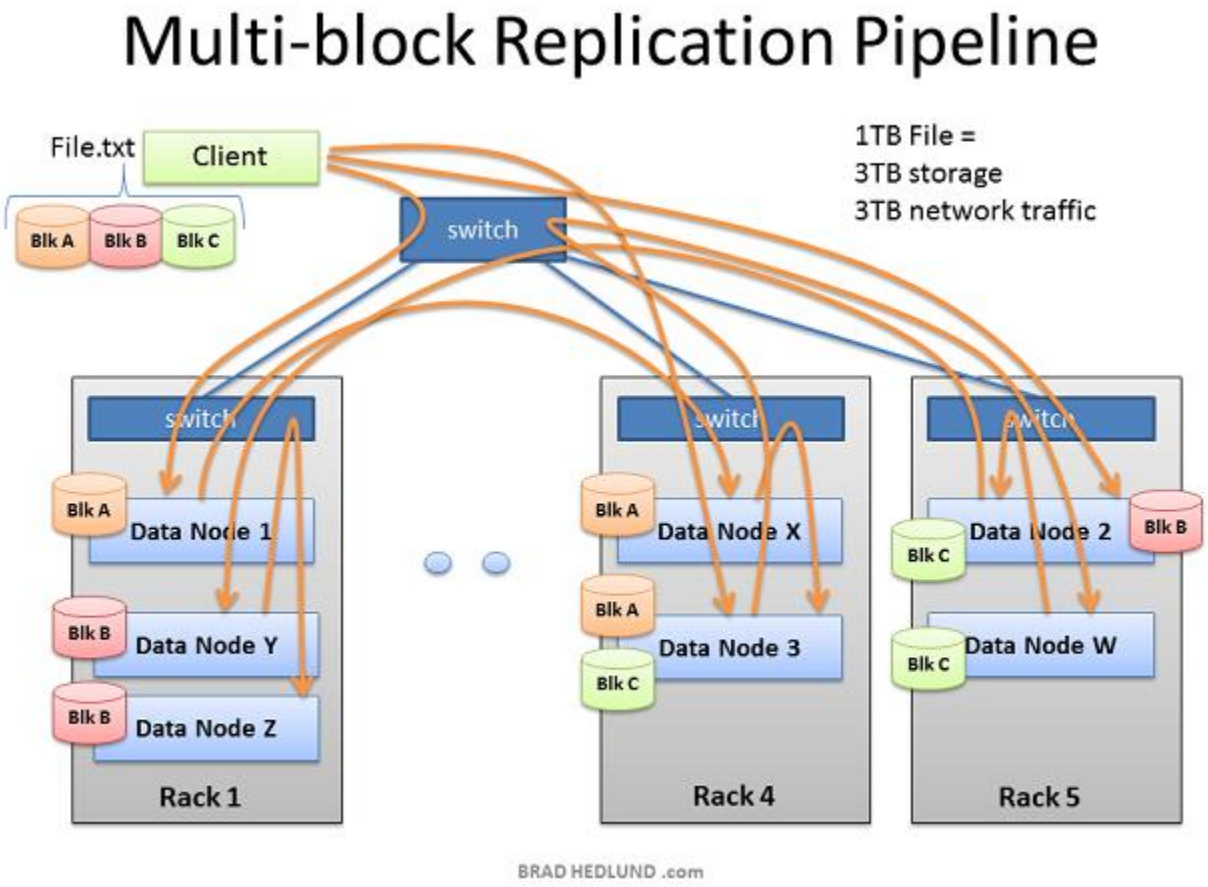
1) Symptom-based fault detection

- The same symptoms might manifest themselves during an intrusion

- Symptom-based fault-detection mechanism could be adapted to serve as a provisional intrusion detector

2) Replication of data - Hadoop

- Hadoop triplicates data for reliability as well as access time optimization

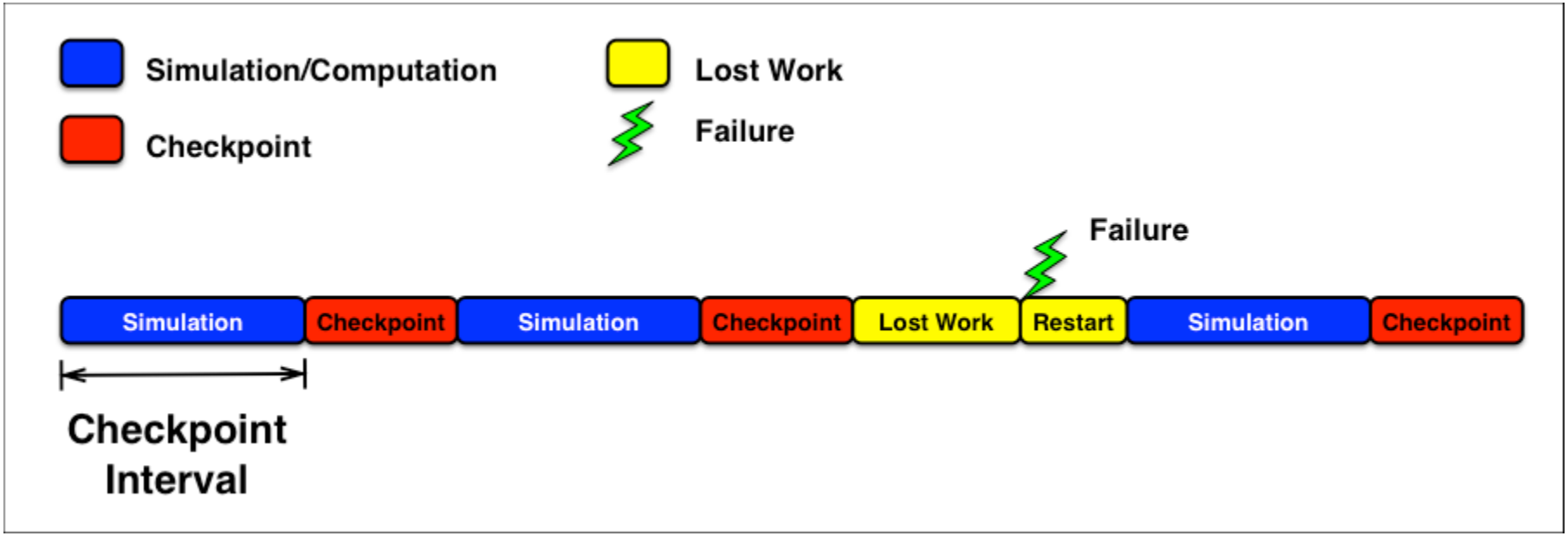


2) Replication of data - Hadoop

- Replication of data -> normally increases the attack vectors
- However replicated data benefits from geographical distribution
- Can help recover from malicious data corruption if at least one copy preserves data integrity

3) Checkpointing

- Checkpoints: typically used to recover from transient errors



- Could be useful to restart and undo computation in case of a integrity compromise

3) Checkpointing

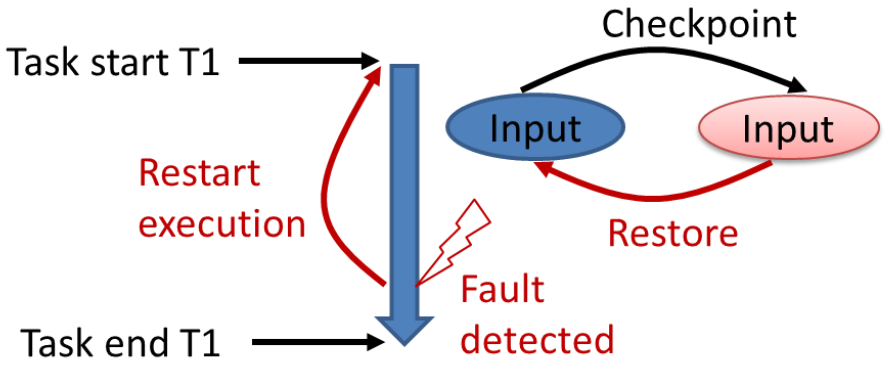
- On the other hand Checkpointing can be a security risk
  - Snapshot of complete system state – can include the OS

- The less critical state exposed the better

- System(OS-kernel) vs. runtime vs. application level ckpt
- Coordinated vs. asynchronous ckpt
- Monolithic vs. incremental ckpt
- Single vs. multi-level ckpt

4) Fault containment

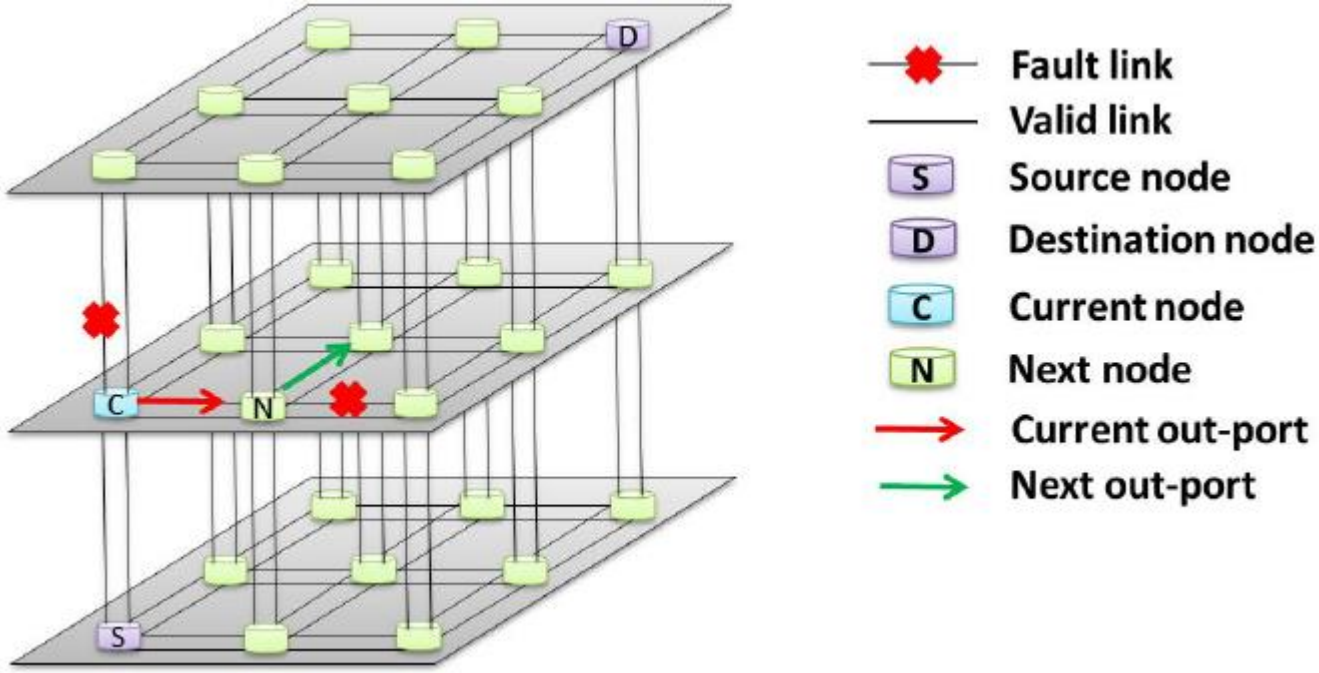
Recover task execution from the detected errors.
Contain errors within the task boundaries.



Inputs are known at runtime through programmer annotations.
Overheads of checkpointing and recovery are minimal.
Recovery is asynchronous.

- Subasi et al., NanoCheckpoints: A Task-based Asynchronous Dataflow Framework for Efficient and Scalable Checkpoint/Restart, PDP2015
- Task-based programming models could be useful for containing the damage from security violations, help build security sandboxes

5) Fault-tolerant routing



- Fault-tolerant routing: can help (not only in times of fault), but by
  - Bypassing in case a certain router is compromised.
  - Taking different paths for different messages to minimize the impact of a security violation

6) Execution replication

- Can help to identify attacks (by identifying divergent execution)
  - Multithreading-based reliability: copies of threads on SMT, on different cores
  - S/390 G5 processor lockstepping
  - N-way redundancy

- However, there is a catch: N-version programming* can increase security vulnerability – attacker can exploit the weakest link
  - Allocation in C
  - Memory model in Java

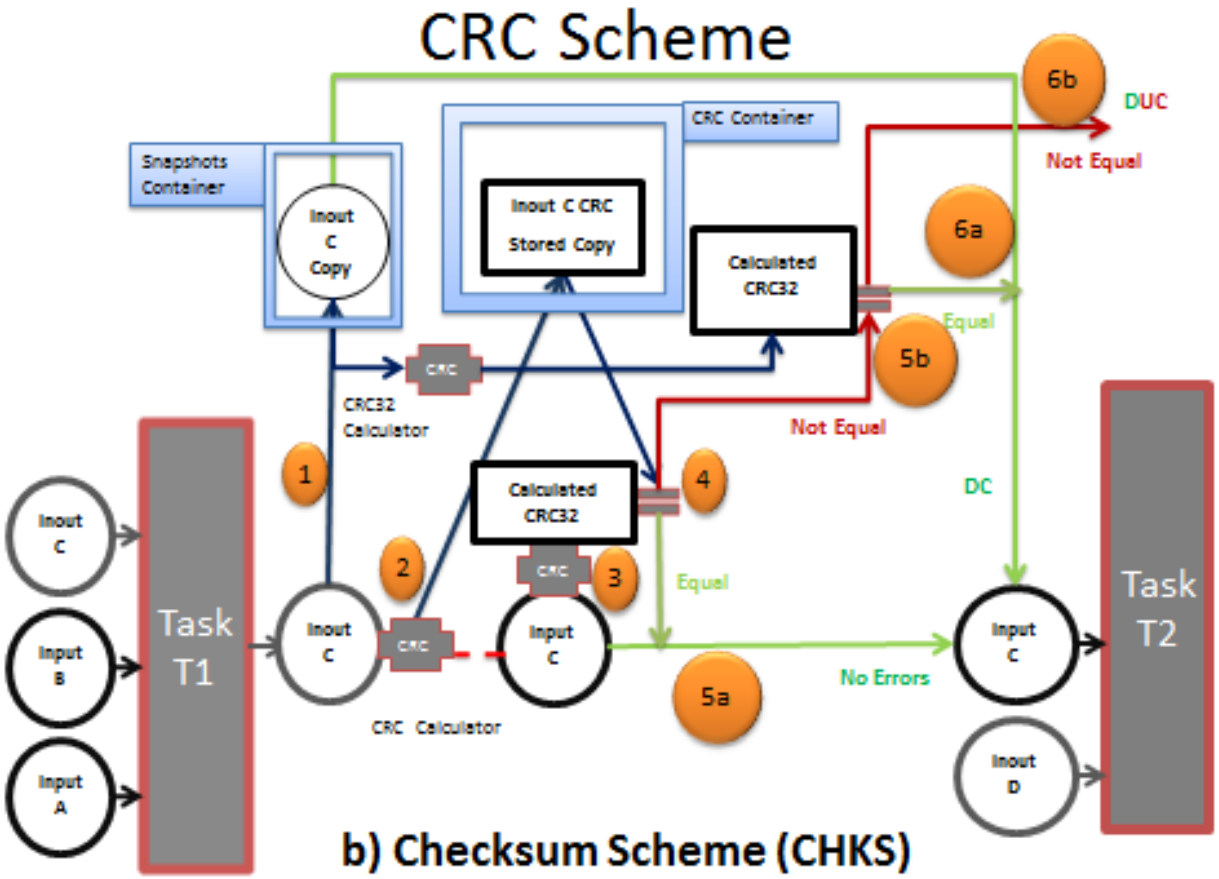
* Thanks to Vassilis Prevelakis for the suggestion

7) Coding

- SW-based CRC for detecting/recovering from memory errors

- Leverages CRC Instruction in X86, ARM

- Could be useful for detecting unauthorized memory tampering



Subasi et al, CRC-based Memory Reliability for Task-parallel HPC Applications, IPDPS 2016

8) Fault prediction

- At runtime: monitor execution, primary outputs, use machine learning to detect deviations
- Gainaru et al., Failure prediction for HPC systems and applications: current situation and open issues, SAGE 2013
- Adopt machine learning based failure prediction for detecting intrusion
  - Deviations not necessarily linear (SVM)
  - Dos not need a “symptom” to be triggered

Security → Fault Tolerance Random ISAs

- Randomized ISA support, also for power
  - Carver, Microprocessors for root-of-trust, Hotchips 2013

- Intermittent Faults might be masked
  - Use randomized ISA support to recover from Intermittent Faults

Fault-tolerance mechanisms

- Replication of Data
- Replication of Execution (in time, or in space)
  - Duplex
  - Triplex – Majority Voting
- Checkpointing
  - System/application, coordinated/asynchronous, monolithic/incremental, single/multilevel
- Coding – Parity, ECC, CRC, RAID, Residue

Fault-tolerance mechanisms

- Error detection
- Error correction
- Fault prediction
- Fault containment


BSC
Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación
THANKS