Data Analytics to detect corruptions in numerical simulation

Data

Systematic & non systematic corruptions

Attack(s)
Bug(s)

Radiation

Application

Numerical lib.

Runtime(s)

OS

Hardware

How to improve the **trust** on the product of the execution?

Related to the problem of numerical simulation integrity

Examples:
- BG/L L1 cache
- FERMI GPUs at LANL
- Kepler GPUs
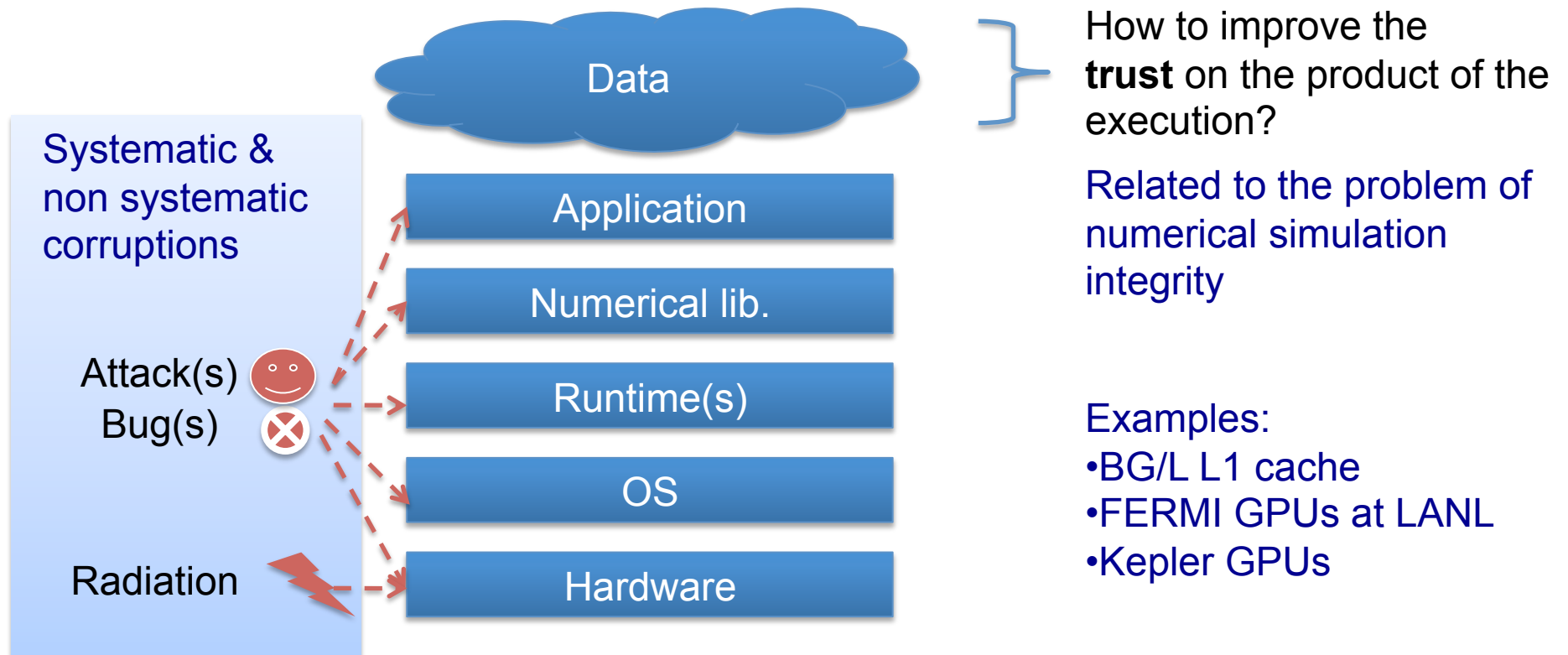
Franck Cappello, Leonardo Bautista Gomez, Sheng Di, ANL

U.S. DEPARTMENT OF ENERGY

Constraints & approach

Non systematic SDCs:
Specific: ABFT
Generic: replication / comparison

Systematic SDCs:
-Replication using the same execution configuration does not help
-N-version and Recovery blocks [2] not applicable in HPC (require implementing multiple versions of complex codes, which will be too expensive.)

→We need an external observer monitoring the product of the execution: data analytics

Runtime monitoring/verification

External Observer

Data

Application

Numerical lib.
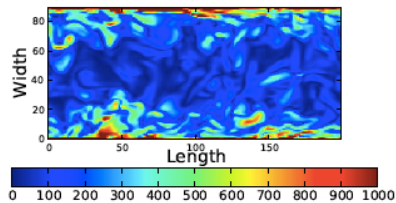
Runtime(s)

OS

Hardware
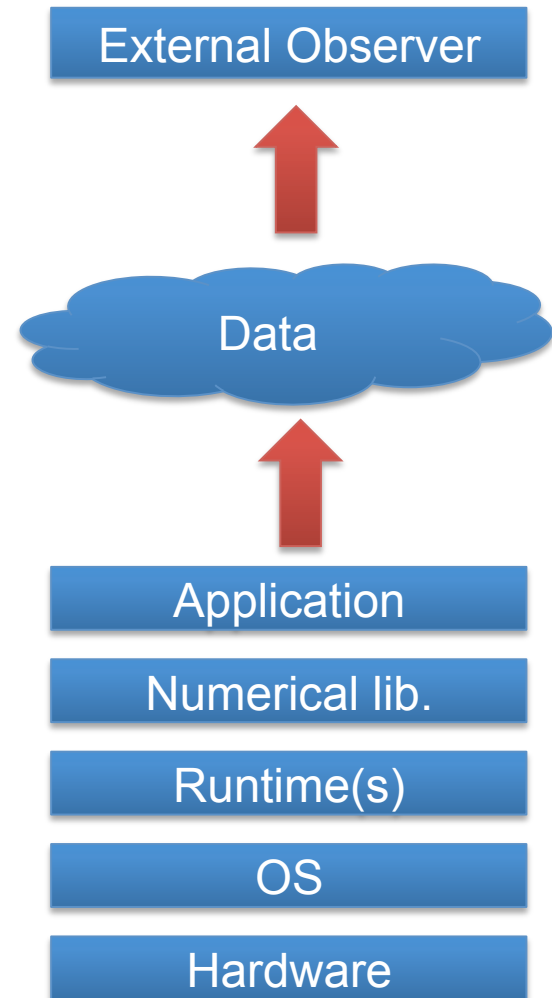
Observation

What property of the data set can be exploited by the external observer?
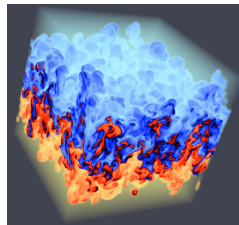
We Looked at applications using iterative methods to compute solutions of initial value problems for ODEs and PDEs
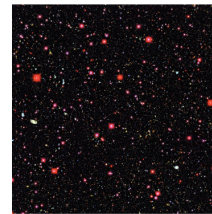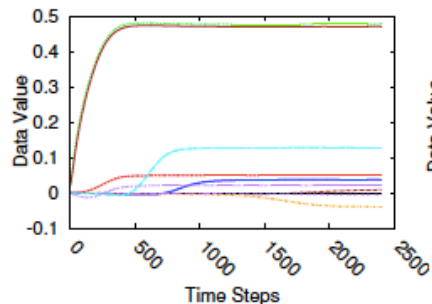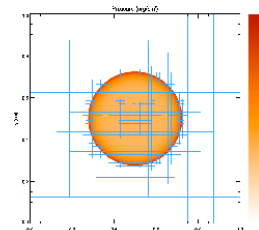
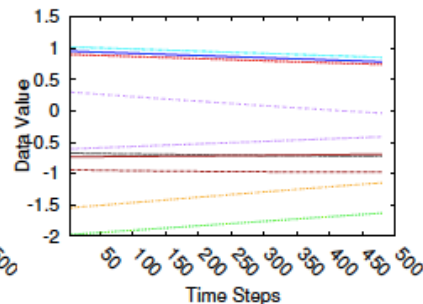Turbulent CFD   Nek5000   HACC   FLASH
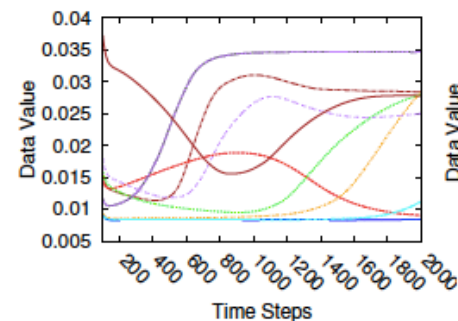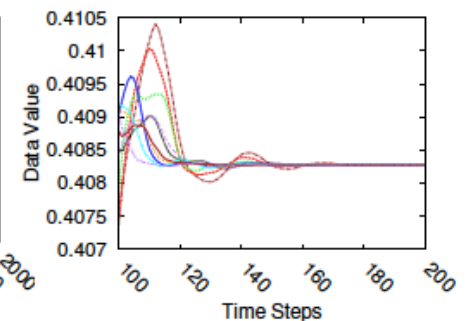

(a) Vorticity in turbulent fluid

(a) Nek5000-Vortex
density

(b) Nek5000-Eddy
pressure

(c) FLASH-Sedov
density

(d) FLASH-Sod
density

→We have observed that for many simulations (production codes and data sets), application data evolves in a rather smooth way across time steps.
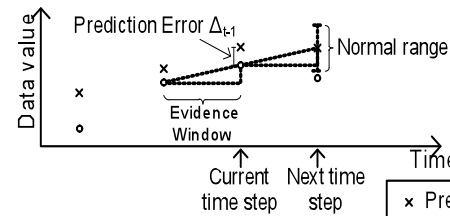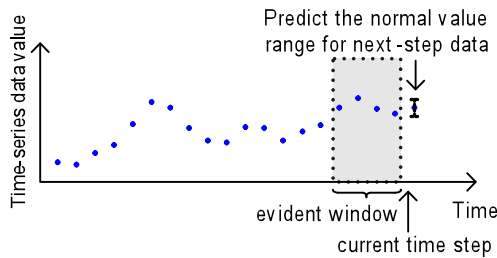→We are NOT claiming that ALL applications have this property.
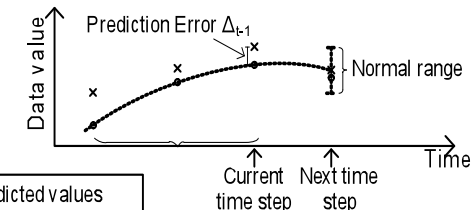
Data analytics based SDC detection

Second observation: Users accuracy requirements: 10^-6 (HACC), 10-^8 Nek5000

Focus on harmful SDCs (HSDCs):

Obvious? | Negligible?

| 0 | 1 0 0 0 0 0 0 1 | 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 |

+
−
Exponent | Mantissa

Detection Approach: **Model variable trajectory**, **predict the next value, compare**

Predict the normal value range for next-step data
Time-series data value
evident window → Time
current time step

Prediction Error Δ_{t-1}
Data value
Normal range
Evidence Window
Current time step | Next time step | Time
× Predicted values
○ Observed true data
(a) LCF method

Prediction Error Δ_{t-1}
Data value
Normal range
Current time step | Next time step | Time
(b) QCF method

Results:
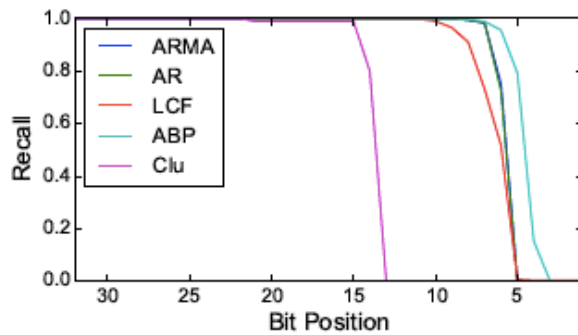
(a) HACC (particles' position) (b) Turbulence-CFD (velocity) (c) Nek5000-vortex
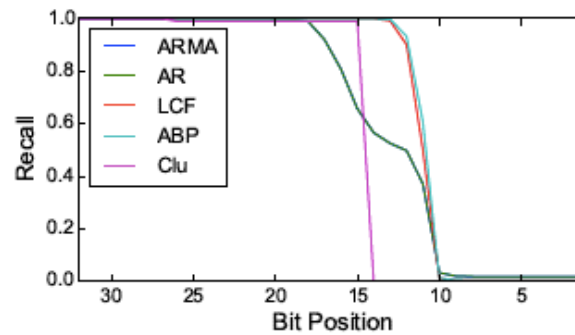
Legend: ARMA, AR, LCF, ABP, Clu — Recall vs Bit Position
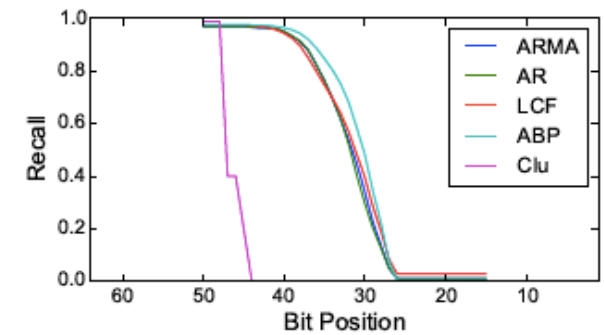
HSDC Detection recall of 99% for HACC and TCFD, 66% for Nek5000 and FLASH
→ **Very promising approach to address this complex problem.** Needs more exploration.