

Breakout 2.1; End-to-end data protection

Scope

- End-to-end means protecting the data between inception and, later, use

Uncertainty Quantification

- Generates a quantified “confidence” in the result
- In DOE/NNSA apps; Run big 6 month simulation, simultaneously run lower fidelity simulations to get a handle on the error present in the large run
- Can we codesign with the UQ folks to incorporate data reliability?
- Ultimately, goal is to “make sure the user is able to get his job done”
 - UQ or end-to-end, two paths to the same end

Standards Based Protection Schemes

- Such as T10DIF, now T10PI
- Need an API for the app, maybe extended attributes?
- Various implementation schemes discussed
 - Distributed checksum generation
- One size fits all solution won't work
 - Data has different value and value changes over time
- Hardware should be reliable!
- We should contemplate unreliable hardware

Accept Reality

- Errors happen; Fess up!
 - Seems dangerous, we plan to barely exceed hardware, hardware plans to barely exceed us, ratchets down to the ridiculous
 - Admitting you are in denial is the first of the twelve steps.
- Would multiple, parallel, paths through the stack help?

The Cost/Value Proposition

- Can it become more expensive to run the job twice than to detect/correct?
- Do it anyway, eventually all data will reside in the cloud
- But... BG is deterministic. Go ahead, run it again and get the same (erroneous) result
- If end-to-end, would apps use it?
 - Researcher is betting their reputation; You bet they would!

Summary

- End-to-end will be important
- Detection is important
- There will be a collection of methods to deal with this
 - Available at various levels
 - Making them play together is a doomed exercise
- Apps need to participate
- UQ may be another, “cheaper” way to reach the same goal