Does Error Resilience Matter in the age of Approximate Computing?

Karthik Pattabiraman,

Bo Fang, Guanpeng Li, Qining Lu, Anna Thomas, and Jiesheng Wei

University of British Columbia (UBC)

http://blogs.ubc.ca/karthik/
What this talk is about

• **Approximate Computing**: Exact results don’t matter (much) - compromise on correctness

• **Error-Resilient Computing**: Can we produce correct results in the presence of hardware faults?

• **Question**: If correctness does not matter, then is it worth the trouble to build error resilient systems?
What this talk is about

Which one are you?

1/2 FULL? 1/2 EMPTY?

Approximate Computing

Error Resilient Computing
Approximate Computing: Myths and Reality

- **Myth 1:** Soft computing applications can tolerate almost all errors in their data.

- **Myth 2:** Crashes are harmless. SDCs or output corruptions are what matter in practice.

- **Myth 3:** Programmers are good at writing correctness checks or annotations in the code.
Approximate Computing: Myths and Reality

• **Myth 1:** Soft computing applications can tolerate almost all errors in their data.

• **Myth 2:** Crashes are harmless. SDCs or output corruptions are what matter in practice.

• **Myth 3:** Programmers are good at writing correctness checks or annotations in the code.
Soft Computing Applications

- Applications in machine learning, multimedia etc.

Original image (left) versus faulty image: JPEG decoder
EDCs: What are they?

- Large or unacceptable deviation in output

EDC image (PSNR 11.37) Vs. Non-EDC image (PSNR 44.79)
Error Resilience: Initial Study

References: DSN’13, SELSE’13, TECS – in press
Approximate Computing: Myths and Reality

• **Myth 1:** Soft computing applications can tolerate almost all errors in their data

• **Myth 2:** Crashes are harmless. SDCs or output corruptions are what matter in practice

• **Myth 3:** Programmers are good at writing correctness checks or annotations in the code
Fail-stop Assumption
Long-latency Crash

Send messages

File I/O

Checkpoints
Checkpoint Corruptions

References: DSN’15, ISSRE’15
Approximate Computing: Myths and Reality

• **Myth 1:** Soft computing applications can tolerate almost all errors in their data

• **Myth 2:** Crashes are harmless. SDCs or output corruptions are what matter in practice

• **Myth 3:** Programmers are good at writing correctness checks or annotations in the code
Programmers are busy!

- Last thing they’re going to worry about is annotating data/code as critical or non-critical

- Writing good correctness checks is hard – many checks are either ineffective or wrong

- Finally, programmers tend to be conservative – Will mark everything as critical if in doubt
Approximate Computing: Myths and Reality

• **Myth 1:** Soft computing applications can tolerate almost all errors in their data

• **Myth 2:** Crashes are harmless. SDCs or output corruptions are what matter in practice

• **Myth 3:** Programmers are good at writing correctness checks or annotations in the code
Error Resilience **DOES** Matter in the age of Approximate Computing!

Karthik Pattabiraman,

Bo Fang, Guanpeng Li, Qining Lu, Anna Thomas, and Jiesheng Wei

University of British Columbia (UBC)

http://blogs.ubc.ca/karthik/