CCC3 Fritz: Programming Model Performance amid Faults

SAND# _____

SAND2013-0737P

Exascale systems will experience errors/faults much more frequently than petascale systems*

*Towards Exascale Resilience, Cappello et al., Intl. Journal of High Performance Computing Applications Nov 2009 vol. 23 no. 4 374-388
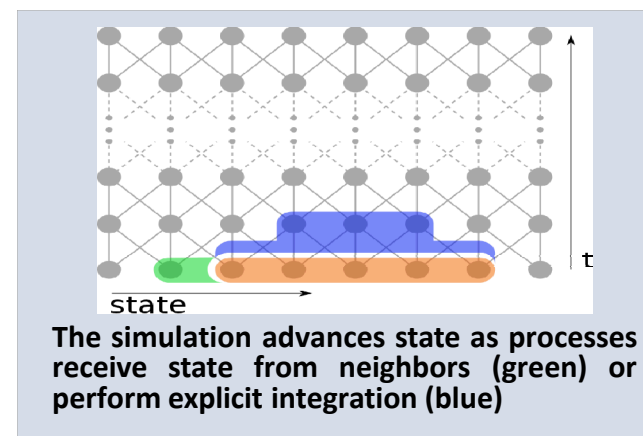
- **Cause:** There is a significant increase in the number of components with insufficient improvements in mean time to failure for each one.

- **Our goal:** Discover the right approach for fault-resilient programming on extreme-scale systems.

PI: Nicole Slattengren
Code: Cellular Automaton
Run time: 1,433,762 cpu hours

CCC3 Fritz: Programming Model Performance amid Faults

- In this work we quantified performance & fault resilience characteristics of an asynchronous many-task programming model

- We explored the propagation of delays due to failures through a 1D cellular automaton code

  - State of a cell at step k depends on the state of that cell and its two neighbors at step k-1

  - Fault-induced delays were injected using a Poisson fault model

 - We measured the time until all tasks had completed ("maximum completion time")

The simulation advances state as processes receive state from neighbors (green) or perform explicit integration (blue)

Milestone supported: Scalable, Fault-Resilient Programming Models entry WBS 1.5.4.4 in the ASC FY12 Implementation Plan (IP)

CCC3 Fritz: Programming Model Performance amid Faults

- The asynchronous task-driven automaton code scales up to 50,000 cores on Cielo
  - Delays do not compound each other at scale
  - Local check-pointing strategies show promise
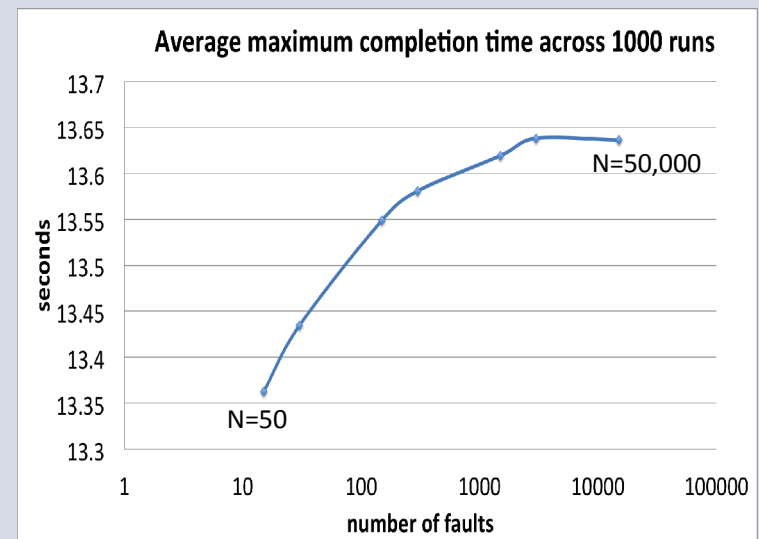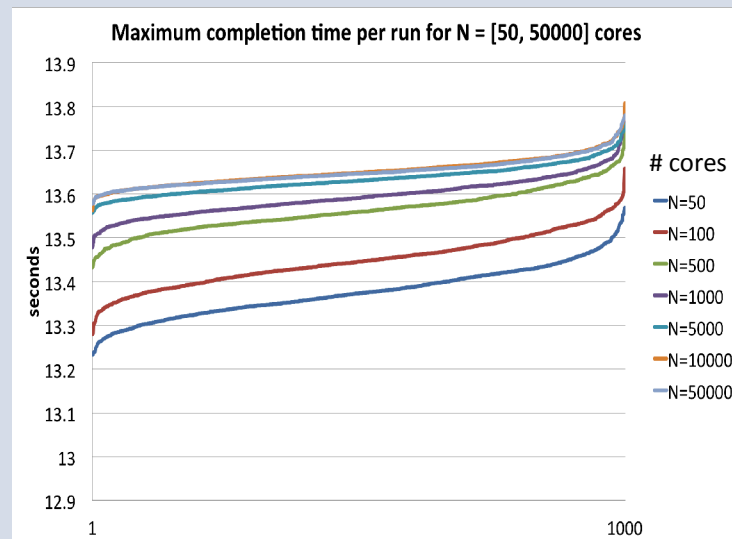
Run number (sorted by maximum completion time)

Average over runs

Poisson fault model: # of faults increases with number of cores