Resilience Panel

Overview and Questions

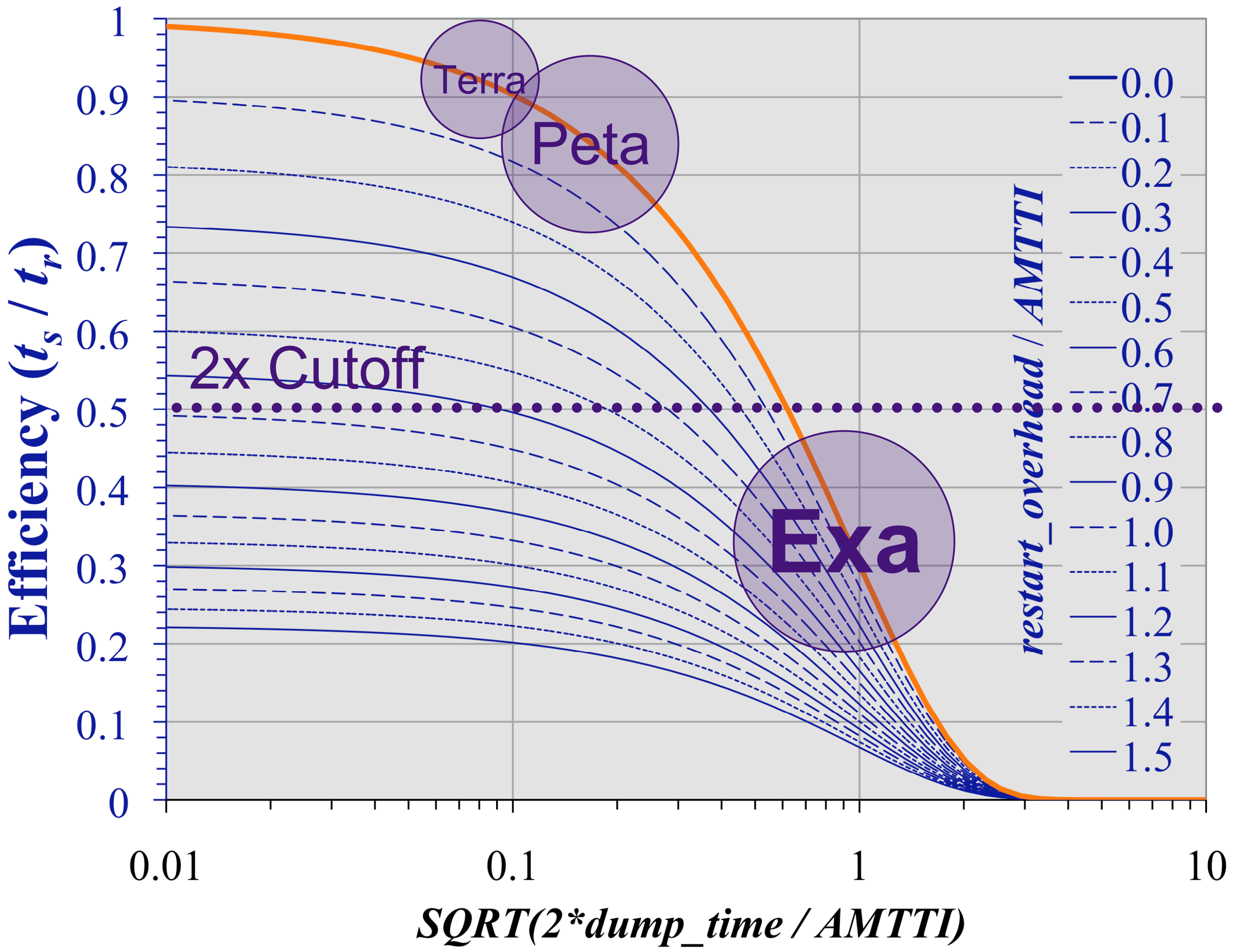
John T. Daly

ACS / CEC

Three challenges to traditional fault-tolerance approaches

- Increasing component counts and densities result in increasing numbers of permanent, intermittent and transient system faults
- Redundancy and replication based fault-tolerance schemes increase the number of system resources dedicated to fault recovery
- Proliferation of dependent and silent failure modes reduce effectiveness of monitoring to detect faults that impede application progress

Recovery based fault-tolerance approaching a theoretical limit



Resilience is an application-centric computing paradigm

- **Definition**: Resilience is keeping applications running to a correct solution in a timely and efficient manner in the presence of system degradations and failures
  - correct solution ⇒ protecting the application from data corruption and Byzantine faults
  - timely and efficient ⇒ consideration of tradeoffs in power, productivity and performance
  - degradations and failures ⇒ any hardware or software event that impedes application progress

How is resilience different than fault-tolerance?

- Fault-tolerance is…
  - Segregated — the system or application must keep itself running in spite of component failure
  - Reactive/Fatalistic — redundancy and replication are used to recover from failure
- Resilience is…
  - Integrated — the system works with applications to keep them running in spite of component failure
  - Proactive/Optimistic — monitoring, analysis and response are used to circumvent failure

Questions for the Panel

… or anyone else who knows the answers

System available cycles vs. application usable cycles

Are you concerned about differences between application and system fault accounting?

A. No, I don’t think that this is a problem
B. Yes, but I’m not sure we can do much about it
C. Yes, so we need better system monitoring
D. Yes, so we need system & application monitoring
E. None of the above

Resilience requirements are not the same for everybody

What types of fault-tolerance or resilience are most applicable to your problem space?

A. I’m happy rerunning the problems/parts that fail

B. Recovery oriented computing is all I ever need

C. I’d like task/data redundancy at compile/runtime

D. Give me full fault prediction and migration or bust

E. None of the above

Correct solutions in a timely and resource efficient manner

What price are you willing to pay to address silent data corruption and Byzantine faults?

A. I’m not concerned about getting the wrong answer
B. I’d be willing to sacrifice up to 20% performance
C. I’d be willing to pay up to 20% more for hardware
D. I’ll run or buy everything twice if I have to
E. None of the above

Next steps towards developing a resilience infrastructure

What are the most important and/or difficult technologies required for resilience?

A. Proper instrumentation and analysis for detection and prediction of system faults

B. Reporting mechanisms and metrics for accurately describing and disseminating failure information

C. Dynamic, real time reconfiguration of running jobs

D. Prevention of undetected errors in logic or data

E. None of the above