Reliability in High Performance Computing:

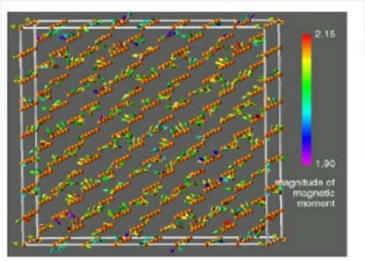
peanuts or hot potato?

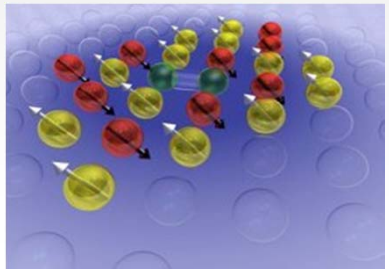
Ramon Canal

Dept. of Computer Architecture

Universitat Politècnica de Catalunya

Motivation: The Exascale System





?

> 100 Millions cores

8 proc Cray YMP

1500 proc Cray T3E

180 Kcores Cray XT5

1988	1998	2008	2018?
1st sustained GFlops award	1st sustained TFlops award	1st sustained PFlops award	1st sustained ExaFlops award

Gordon Bell Performance Award

Motivation – Inside an Exascale System

Datacenter: 10^9 threads

Rack: 10^4-10^5 threads

Socket/blade: 500-5000 threads

Die: 100-1000 threads

Core/tile: 1-10 threads

Blue Waters case:

Petaflop machine

250 errors/h (CPU+MEM)

99,997% recovered

IBM Z-series case:

DMR (lockstep at instruction level)

30% chip real state for reliability and recovery

Motivation – Scale and Failures

- Mean Time Between Failure (MBTF):
  - for supercomputers we talk about MTBF in days/weeks
  - If it happens, we restart the application

- ExaScale MTBF
  - We will have smaller components
  - We will have a million times more components

MTBF of minutes/seconds ?!

Predicted overhead:

up to 30 minutes per checkpoint

at 1 Terabyte/second

Motivation – Technology roadmap (ITRS)

- Problems of extremely small scale
  - Around 11 nm by 2018
- Heat flux and temperature variability over space and time
- Aggressive Frequency switching
- Alpha particles and cosmic rays hitting silicon, causing bit flips
- Physical wearout

Additional correctness checks would **increase power consumption by 15-20%.**

[Dongarra et al., The International Exascale Software Project Roadmap, 2011]

State-of-the art

- Hardware:
  - ECC, DMR, etc.
  - Checksums for network/IO
- OS
  - Rollback-recovery (checkpointing)
  - Proactive (i.e. ontesting)
  - Replication
- Application/Language
  - Nothing in real systems
  - Critical sections

Is it enough?... NO

Recall: MTBF of minutes

System Availability

- For 90% availability with 1M nodes, each node needs:
  - –7 nines without redundancy
  - –4 nines for DMR
  - –3 nines for TMR

$$A = \frac{MTTF}{MTTF + MTTR} = \frac{1}{1 + \frac{MTTR}{MTTF}}$$

9s	Availability	Annual Downtime
1	90%	36 days, 12 hours
2	99%	87 hours, 36 minutes
3	99.9%	8 hours, 45.6 minutes
4	99.99%	52 minutes, 33.6 seconds
5	99.999%	5 minutes, 15.4 seconds
6	99.9999%	31.5 seconds

Not even TMR is enough ☹

Solutions – what's next?

- "Game of Thrones" style
  - Circuit, architecture, ISA, application, OS fight the battle on each one.
  - Possibly overshooting ⇨↑overheads, ↓performance, ↑costs

Solutions – what's next?

- "UN" style
  - –Cooperative approaches
    - –with (traditionally) uncooperative communities
  - –Need coordination efforts from the first day

Know the system impact of circuit/technology/… decisions

Cooperative approach

- Early stage system evaluation can:
  - Drive research/development efforts
  - Reduce Time to Market (TTM)
  - Provide a holistic analysis (MIPS,W,Pfail)

  - Iterative process to converge to optimal solution

  - Must define interface between levels
    - Key to enable smooth interaction
    - Plug'n-play system evaluation possible

Conclusions

- Technology reliability is now felt at the system level
- No level alone can meet the power/performance/goals
- Need cooperation to find the optimal design point
- Early stage variability estimation can help evaluate the impact of each decision on the final system

Reliability in High Performance Computing:

peanuts or hot potato?

Ramon Canal

Dept. of Computer Architecture

Universitat Politècnica de Catalunya