

# ACL Extreme Linux Clusters

---

Pete Beckman, Susan Coghlan,  
Ron Minnich

# Successful Operation

---

User codes running without noticeable  
interrupts



# Testing

- What is the point?
- Can you ever really tell if everything is ok?
- What is the hard stuff?
  - What isn't?

# The hard stuff

- Hardware - you'd think it was easy, and yet...
- Software - so many possibilities, so little time
- Complexity - subtle problems, little changes => big changes, weird interactions: memory errors -> amd failures
- Scale - sheer number of things that can go wrong



# A few key points

- Commodity - are the integrating vendors up to the challenge?
- It has to be a science, not an art
- If it's done right, it can be used to feed info to the system that automatically handles problems.

# What we do now

- correctness:
  - Basic tests - simple tests of the interconnect, simple checks for node correctness, simple mpi tests
  - A little bit more - simple apps that run quickly and have known output
  - Going further - complex apps that test lots of components - NAS PB, Linac
  - QCD - super app, able to leap tall buildings in a single bound



# What we do now, cont....

- performance:
  - Basic tests - benchmarks of components - ping-pong, streams, etc.
  - Getting there - "real" benchmarks, is there one true benchmark?
  - Real stuff - user codes, gasp.

# Test Framework

- test \*everything\*, not just the obvious
- consistency in the framework is important
- output must be succinct
- mix of apps
- if things look good today, will they look good tomorrow?
- regression testing is necessary

# Reliability

- at this scale MTBF becomes essentially meaningless
  - hardware will fail, and lots of it
- the real question is how to deal with failures at the user code level
- and just how reliable is that OS, anyway?
- something must handle failures, and with 6000 nodes it has to be automated

# Availability

- not just hardware downtime
- cycles used for monitoring, testing, etc.
- all the pieces have to be there or the system isn't really available
- Utilization - the correct measure?

# Servicability

- can we drive the cost up to the level of the old supercomputers?
- moving from “desktop” to “server” already a good start
- hot swappable everything?

# Where do we go from here?

- as usual, integration is the issue
- we have to take it from warm fuzzy to real data
- what Bill said - M,M,M D,D,D