

Can Petaflops Computing Survive System Software?

Al Geist

Oak Ridge National Laboratory

Petaflops II

February 15-19, 1999

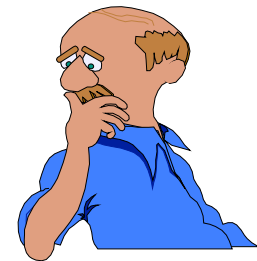
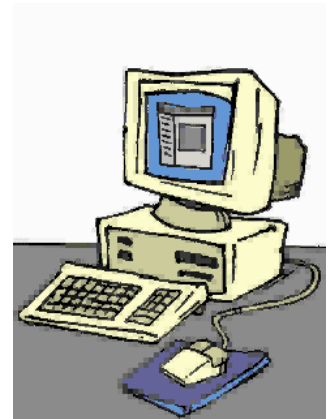
Santa Barbara, CA

How to Break the Speed of Light

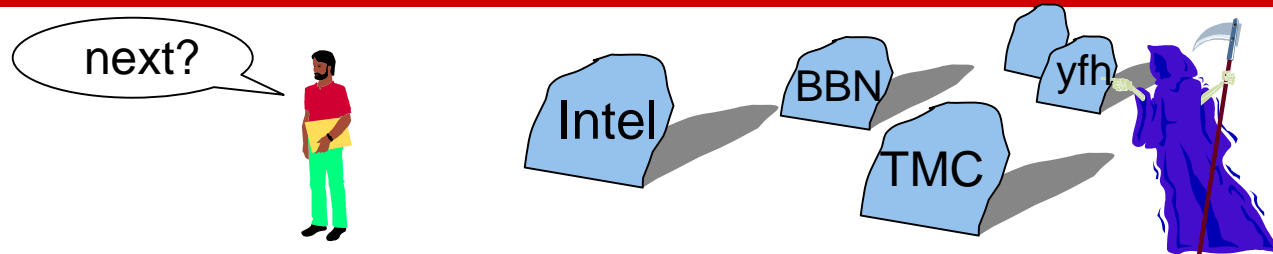
- ◆ Speed is not a problem if the answer doesn't have to be right.
 - Do you care? Numerical accuracy
 - Do you know? Validation issues
- ◆ Scientists seem overly concerned about getting the right answer.
 - Concern for their reputation and integrity
 - Some even feel the answer is the product
- ◆ Such is not the case for those who report how fast their computers are.

Fault Tolerance

- ◆ Mean time to failure of 128,000 cpu system measured in minutes.
 - problem when MTTF is $O(\text{synch time})$
 - less than application startup time
- ◆ How long do I wait
 - to find out if something has gone wrong?
 - memory is a million cycles away.
- ◆ Validation of applications – when...
 - it has subtle synchronization error at $p > 78,347$
 - HW failure loses my asynchronous message



Applications outlive Hardware



- ◆ Programming paradigms are not going to change much.
 - ⑧ Portability is important to developers.
- ◆ Applications are developed and used over several decades.
- ◆ In 2007 Applications will still be written in Fortran.
- ◆ Who is going to write a (Fortran) compiler for HTMT?

Need for Adaptive System Software

- ◆ **KISS petaflop system software—don't need 100,000 copies of Unix/Windows2005**
 - ⑧ Dynamically configure environment to app needs
 - ⑧ Less to break, less to watch
- ◆ **Needs to automatically detect and adapt to “changes” in the system. Note problems happen at petaflop speeds!**
 - ⑧ Cost of hardware support for detection
 - ⑧ Migration of tasks away from bad spots.
 - ⑧ Reroute messages around failures.
- ◆ **Distributed Control for fault tolerance**