The What of Parallelism?

- Solve bigger problems (weak scaling)
  - Largest direct numerical solutions have evolved from meshes of $1024^3$ to $8096^3$ in the last ten years
  - Reveals fundamental structure in the nature of turbulent flow

- Solve the same problems faster (strong scaling)
  - Brute force a 56-bit DES key has evolved from
    - $20M (est) 1976$
    - 1998, 4.5 days, $250,000$
    - 1999 22 hours, $250,000$
    - 2006, 9 days, $10,000$
    - 2008, 1 days, $10,000$
The Why of Parallelism?

- Solve bigger problems
- Solve the same problems faster or cheaper

- Architectures demand it
  - Multi-core
  - GPUs

- Minimize energy consumption
- Maximize investment
Speedup

- The fundamental concept in parallelism
  - \( T(1) \) = time to execute task on a single resource
  - \( T(n) \) = time to execute task on \( n \) resources
  - Speedup = \( T(1)/T(n) \)

http://web.eecs.utk.edu/~huangj/hpc/hpc_intro.php

Parallel Efficiency

- Companion concept to speedup
  - Efficiency = $S(n)/n = T(1)/nT(n)$
  - Informally: fraction of possible performance realized

Strong versus Weak Scaling

- Strong scaling: how the solution time varies with the number of processors for a fixed total problem size
  - Efficiency and speedup described for strong scaling
  - Amdahl’s law addresses strong scaling

- Weak scaling: how the solution time varies with the number of processors for a fixed problem size per processor
  - There is another law—Gustavon’s Law—that governs weak scaling
  - We’ll do weak scaling another day