Scaling

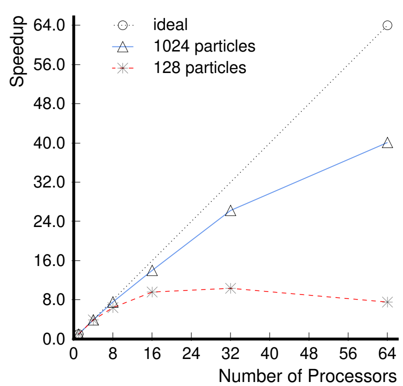
Why is Scaling Important?

- “Speedup” is fundamentally a question of how performance changes as a system parameter is scaled: the number of processors, P.
- Want to solve new problems on new, more powerful machines, not just the same old ones.
  - Need to consider how the problem scales: data set size, algorithmic complexity
- Changes due to scaling can qualitatively alter the load the application places on the machine
- Scientific apps simulate a physical phenomenum to some degree of accuracy; this provides a natural, yet non-trivial scaling.

Speedup revisited

•Fixed problem size is limited in evaluating larger machines

•Scientists want to run larger problems on faster machines



Questions in Scaling

•How much to scale the application?

- Constant Problem Size (CPS)
  - PS(p) = PS(1)
  - want to run larger problems on larger machines
- Memory Constrained (MC)
  - PS(p) = p*PS(1)
  - pretty speedup curve, but execution time explodes
    - O(n^2) problem on O(n) data
- Time Constrained (TC)
  - $PS(p) = \underset{s}{\mathrm{MAX}}\, T_s(p) = T_1(PS(1))$

How should the application be scaled?

- Scaling only data set size is inappropriate
- Should scale under *application* considerations.
- All sources of error should be scaled so that their error contributions are about equal.

Example: A Barnes-Hut Galaxy Simulation

- Different parameters govern different sources of error:
  - Number of bodies (n)
  - Time-step resolution (Δt)
  - Force-calculation accuracy (θ)
- Scaling rule:

<u>All components of simulation error should scale at same rate</u>

- Result: If n scales by a factor of s
  - Δt must scale by factor of $\frac{1}{\sqrt[4]{s}}$
  - θ must scale by factor of $\frac{1}{\sqrt[4]{s}}$

Effects of Equal Error Scaling Rule

If number of processors (p) is scaled by a factor of k . . .

Under Memory Constrained scaling:

- Naive rule => small increase in exec. time
- Realistic tule => increase by factor of $\kappa^{3/4}$

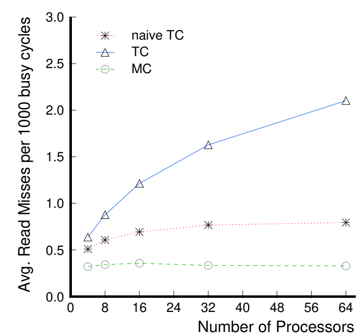
Under Time Constrained scaling:

- Naive rule => n increases by almost k
- Increase in number of particles is less than $\sqrt{k}$

- Time Constrained is most realistic for this application

Scaling of Communication to Computation Ratio

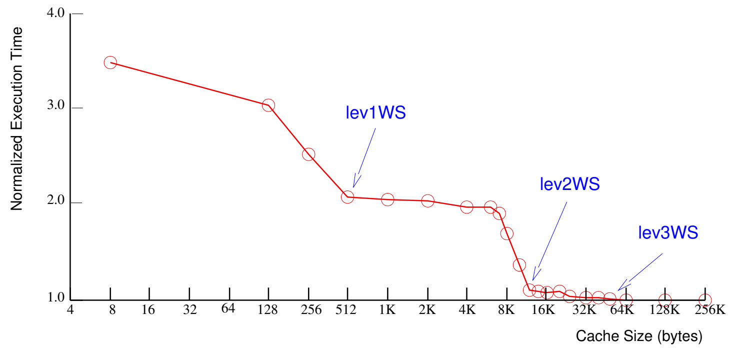
Time and Memory Constrained



- Use of realistic scaling is important
- Ratio increases under TC scaling

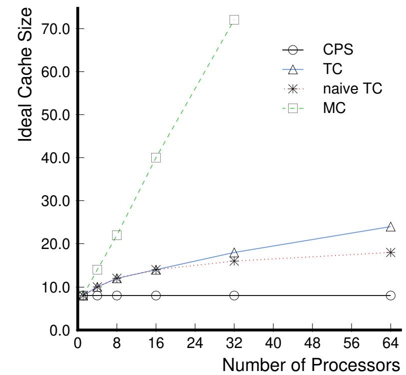
Working Sets

Constant Problem Size



- Knees in curve delineate working sets
- Lev2WS most important to performance

Scaling of Working Set Size



- Use of realistic scaling is important
- Important working set size increases under TC scaling