Supercomputing in Plain English

Tuning

BLUE WATERS

Blue Waters Undergraduate Petascale Education Program

May 29 – June 10 2011

SHODOR University of Illinois at Urbana-Champaign OU Contra Costa College EARLHAM COLLEGE HAMPTON UNIVERSITY

Outline

- Time to Run = Work/Resources + Overhead
- Real Work
- Algorithmic complexity
- Resources and overhead
- Amdahl's law
- Gustafson's law
- Karp-Flatt Metric
- Isoefficiency

Time To Run = Computation/Resources+Overhead

- How much real, distributable work is there to be done?
- How much computing power is available?
- How much overhead is required to set everything up?

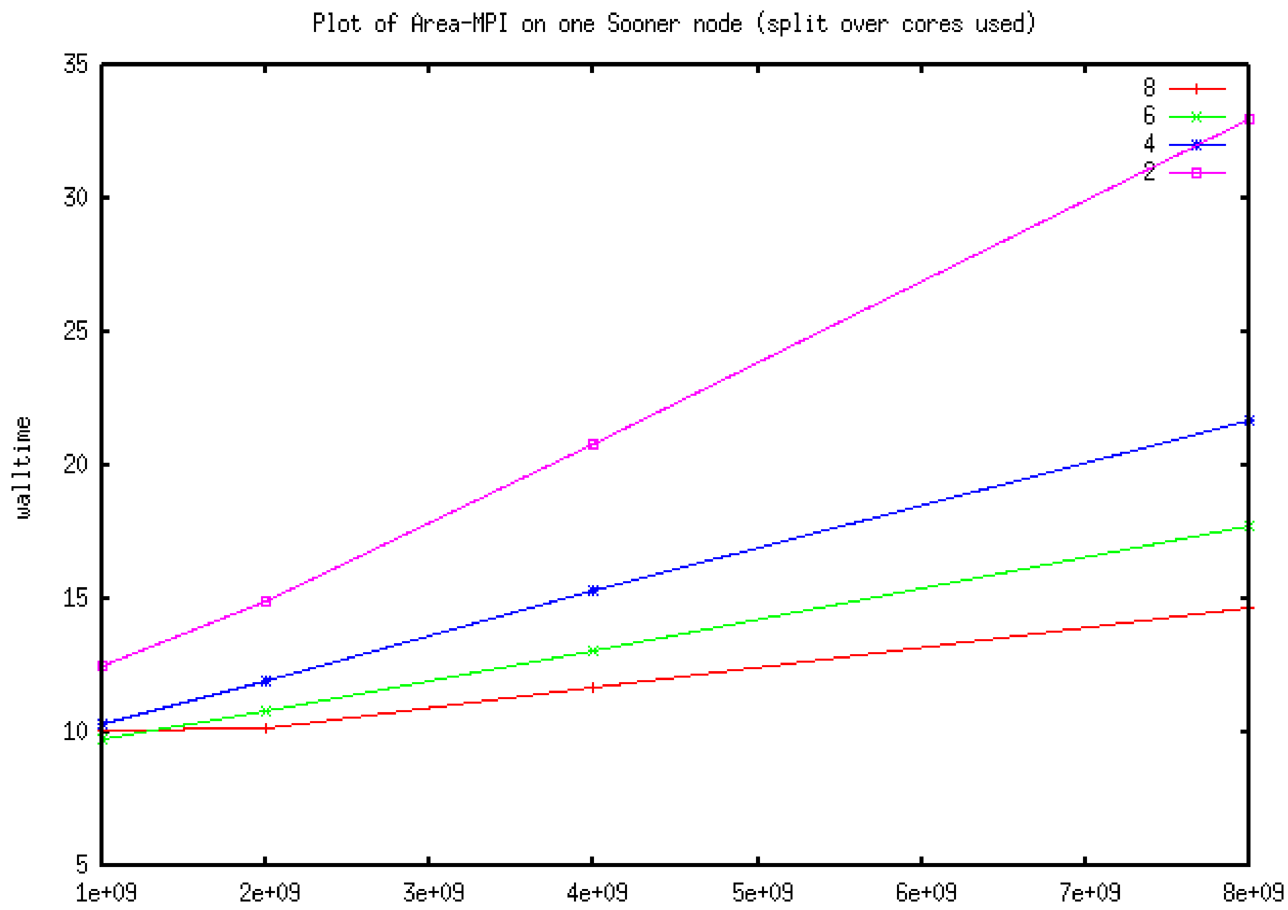
Computation

- The work that actually does the science
- 10% of the code in which 90% of the time is spent
  - This is where real time savings are found
  - Algorithmic complexity

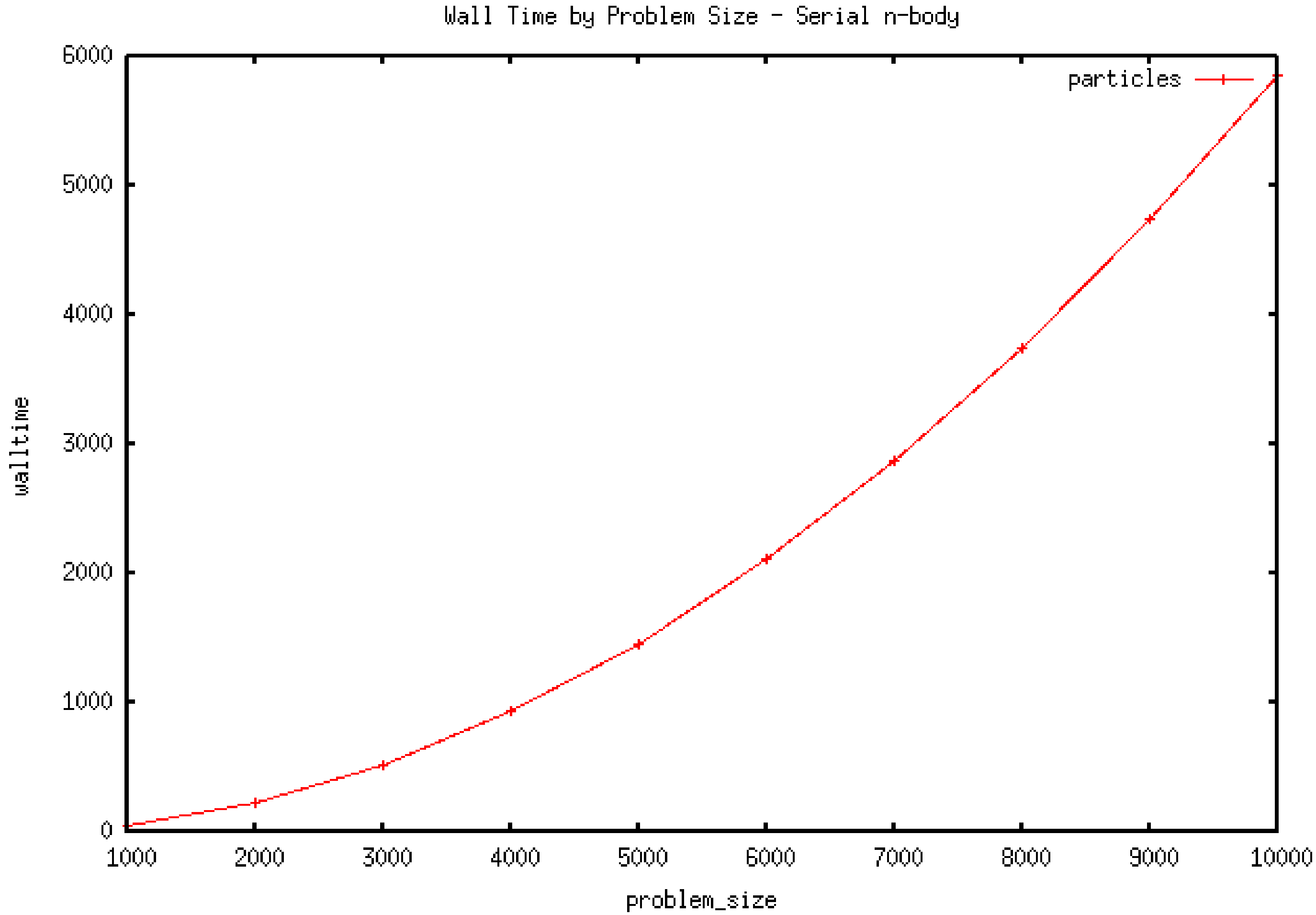
Algorithmic Complexity

- As n approaches infinity, how quickly do the program's requirements (time, memory, etcetera) increase?
- Constant - O(1) – communication overhead in Area Under Curve
- Logarithmic - O(log(n)) - worst case binary search time
- Polynomial
  - Linear - O(n) – computation in Area Under Curve
  - Quadratic – O(n^2) – computation in n-body
- Exponential – O(c^n) – Cryptographic hacking

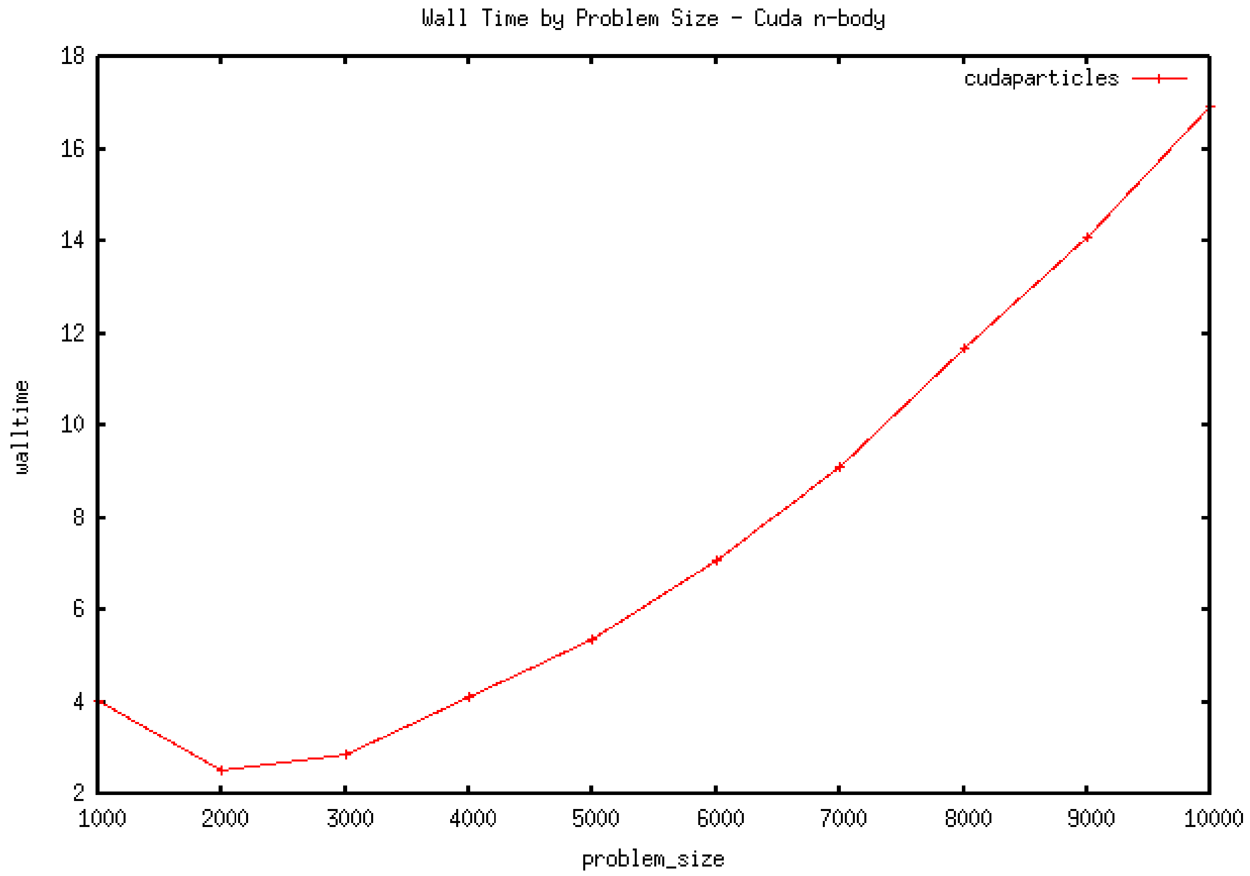
Algorithmic Complexity



Algorithmic Complexity



Algorithmic Complexity



Changing the Algorithm

- function fib(n)

      if n = 0 return 0
      if n = 1 return 1
      return fib(n − 1) + fib(n − 2)

  - Runs in exponential time (O(2^n))

- dictionary m := map(0 → 0, 1 → 1)

  function fib(n)

      if map m does not contain key n
          m[n] := fib(n − 1) + fib(n − 2)
      return m[n]

  - Runs in linear time (O(n))

Time/Space Tradeoff

- Using memory to save time or using time to save memory
- Time is generally what you'll most want to optimize, although many algorithms have to optimize for both time and space.

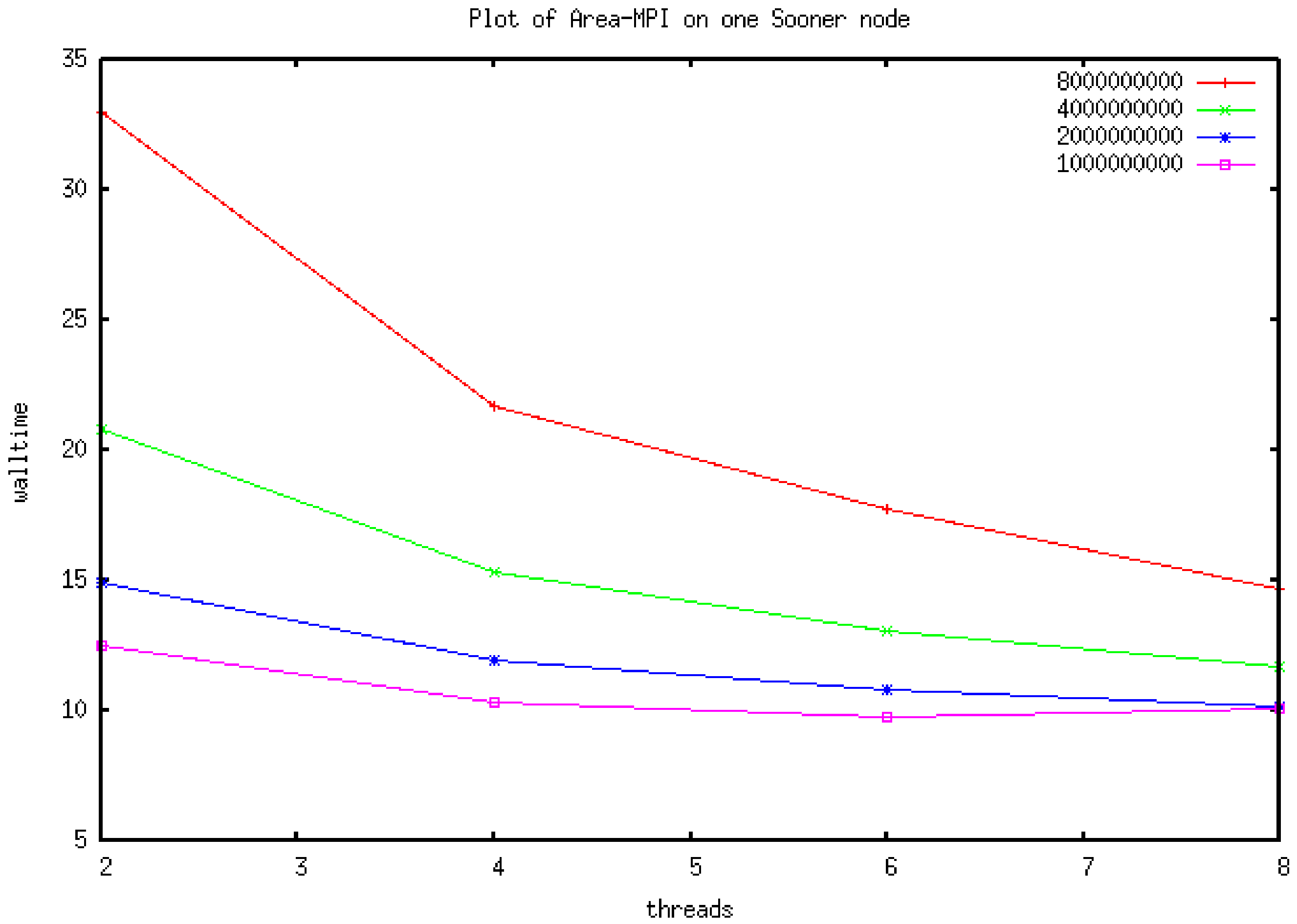
Virtual Memory and Paging

- Using more memory does not slow down the program
  - Until you hit the available memory supply
- Virtual memory uses the hard drive to boost the available memory supply
  - The hard drive is slow, avoid virtual memory
- ulimit -a
  - Shows current limits
  - Virtual Memory will likely not be explicitly limited
- vmstat
- swpd: the amount of virtual memory used.
- free: the amount of idle memory

Resources Available Vs Overhead

- In single-thread programming, the programmer has little control over resources available.
  - Stop other running programs
- Parallelism increases the resources available at the cost of additional overhead in the form of communication.

Resources Available Vs Overhead



Amdahl's Law

- No amount of parallelization can make a program take less time than the time to evaluate its sequential portions.

$$T(p) = T_s + \frac{T_p}{p}$$

Gustafson's Law

- The proportion of the computations that are sequential normally decreases as the problem size increases.

Speedup via parallelization

- Speedup = time for program to run in serial / time for parallelized program to run

Karp-Flatt

A formula for finding e, the sequential fraction of a code

P = number of processors

Ψ= observed speedup

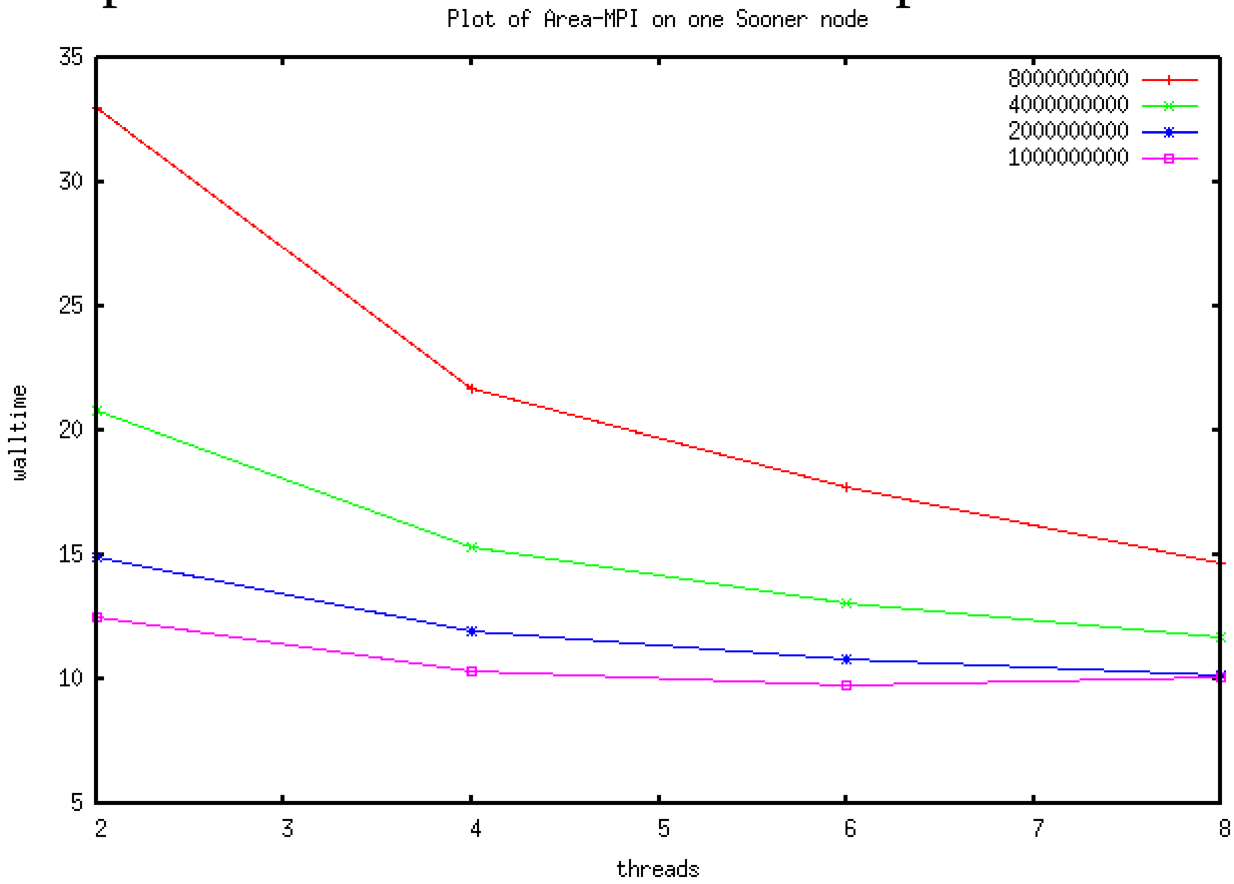
$$e = \frac{\frac{1}{\psi} - \frac{1}{p}}{1 - \frac{1}{p}}$$

Karp-Flatt Paper

http://portal.acm.org/citation.cfm?doid=78607.78614

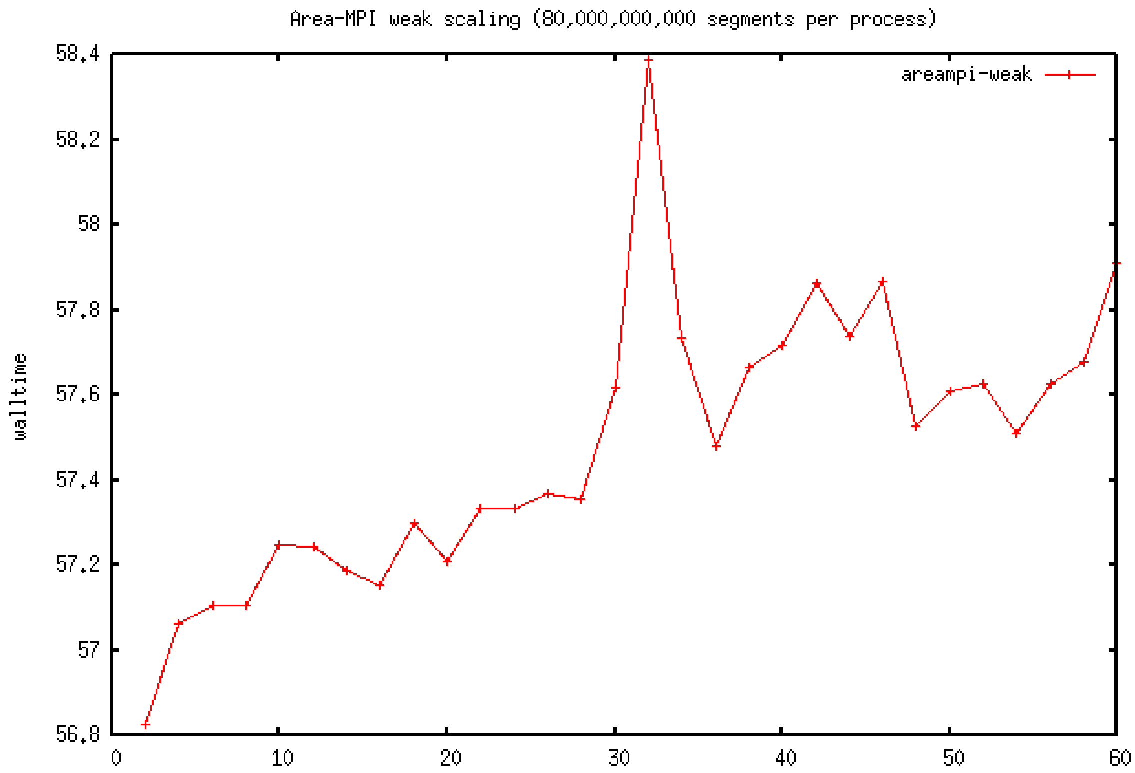
Isoefficiency

Efficiency decreases as the number of processes increases with static problem size

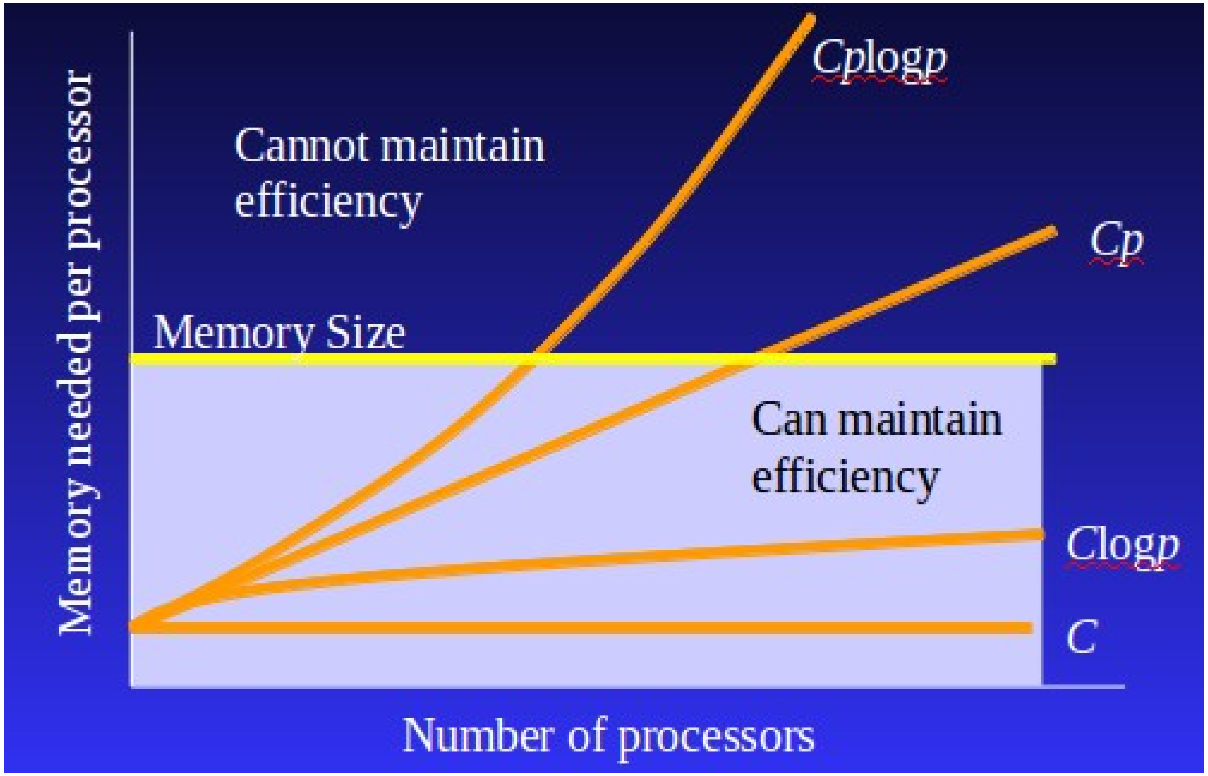


Isoefficiency

How much does the problem size in relation to the number of processes have to increase to maintain efficiency?



Isoefficiency



Isoefficiency paper

http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=00242438