Performance analysis

Jeremy Iverson

College of Saint Benedict & Saint John's University

Performance analysis

- how do we reason about parallel algorithms?
- how can we compare two algorithms and determine which is better?
- how do we measure improvement?

Performance metrics

- execution time (T_p)
- speedup (S)
- efficiency (E)
- cost (C)

Execution time

Serial (T_s)

- time elapsed between beginning and end of execution

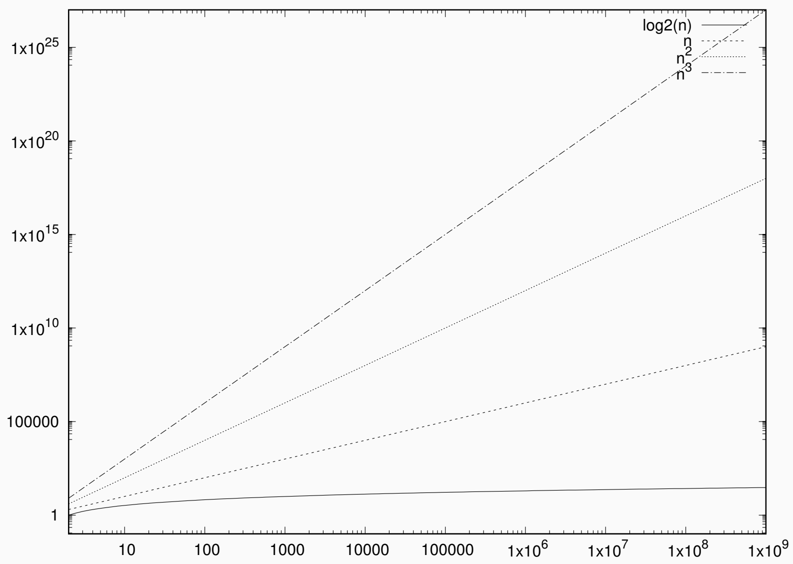
Parallel (T_p)

- time elapsed between beginning of execution and the moment the last processing element finishes execution

- Adding numbers
- Dot-product
- Matrix-vector multiplication
- Matrix-matrix multiplication

1. Adding numbers — $T_s = \theta(n)$ — $T_p = \theta(\log n)$
2. Matrix-vector — $T_s = \theta(n^2)$ — $T_p = \theta(n)$
3. Matrix-matrix — $T_s = \theta(n^3)$ — $T_p = \theta(n^2)$
4. So how long to compute mat-mat for a 10000×10000 mat?

Execution time



Speedup

Speedup ($S = T_s/T_p$)

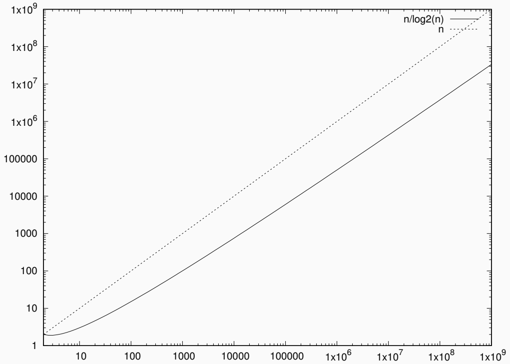
- the ratio of time taken to solve a problem on a single processing element to the time required to solve the same problem on a parallel computer with p processing elements

1. Adding numbers — $S = \theta(n/\log n)$
2. Matrix-vector — $S = \theta(n^2/n) = \theta(n)$
3. Matrix-matrix — $S = \theta(n^3/n^2) = \theta(n)$
4. What are the limits of speedup?

Speedup

Speedup ($S = T_s/T_p$)

- the ratio of time taken to solve a problem on a single processing element to the time required to solve the same problem on a parallel computer with p processing elements



1. Adding numbers — $S = \theta(n/\log n)$
2. Matrix-vector — $S = \theta(n^2/n) = \theta(n)$
3. Matrix-matrix — $S = \theta(n^3/n^2) = \theta(n)$
4. What are the limits of speedup?

Efficiency

Efficiency ($E = S/p$)

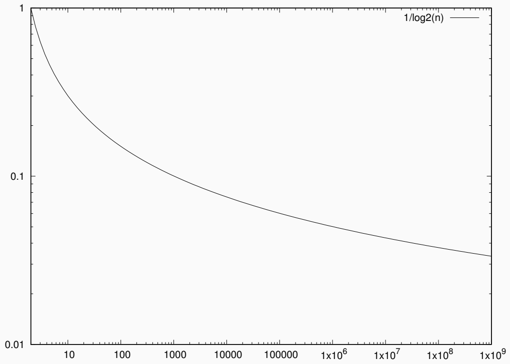
- the ratio of speedup to the number of processing elements — the fraction of time for which a processing element is usefully employed

1. Adding numbers — $E = \theta(n/\log n)/n = 1/\log n$
2. Matrix-vector — $E = \theta(n)/n = 1$
3. Matrix-matrix — $E = \theta(n)/n = 1$
4. Why do you think that the efficiency of adding numbers is less than matrix-vector or matrix-matrix?
5. What are the limits of efficiency?

Efficiency

Efficiency ($E = S/p$)

- the ratio of speedup to the number of processing elements — the fraction of time for which a processing element is usefully employed



1. Adding numbers — $E = \theta(n/\log n)/n = 1/\log n$
2. Matrix-vector — $E = \theta(n)/n = 1$
3. Matrix-matrix — $E = \theta(n)/n = 1$
4. Why do you think that the efficiency of adding numbers is less than matrix-vector or matrix-matrix?
5. What are the limits of efficiency?

Cost

Cost ($C = pT_p$)

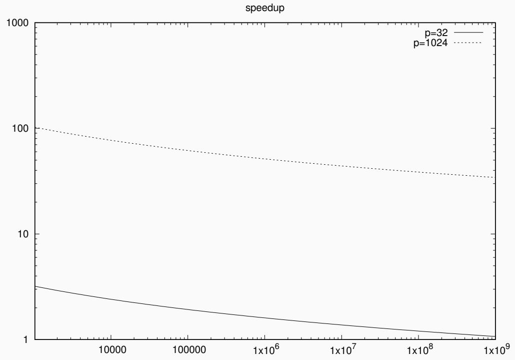
- the sum of the time spent by all processing elements solving the problem
- *cost optimal* if $C = T_s$

1. Adding numbers — $C = n \log n$
2. Matrix-vector — $C = n \times n = n^2$
3. Matrix-matrix — $C = n \times n^2 = n^3$
4. Adding numbers is not cost optimal, the others are

Cost

Cost ($C = pT_p$)

- the sum of the time spent by all processing elements solving the problem
- *cost optimal* if $C = T_s$



1. Adding numbers — $C = n \log n$
2. Matrix-vector — $C = n \times n = n^2$
3. Matrix-matrix — $C = n \times n^2 = n^3$
4. Adding numbers is not cost optimal, the others are