(1)PERFORMANCE ANALYSIS QUINN) a parallel algorithm to decide whether it is worth coding I debugging it. analyze execution time of a parallel programo(i)helps understand the barriers to higher performance

(z) predict the improvement due to increased #procs. AIMS general formula for the speedup achievable by a parallel program performance prediction formulas: (1) Amdahl's law:

(helps decide whether alprogram)

merits parallelization (2) Gustalson-Barsis Paws (evaluate the performance of a)
parallel program (3) Karp- Flatt metric: helps decide whether the principal barrier to speedup is the amount of inherently sequential rode or parallel overhead (4) Isoefficiency metrics

[evaluate the scalability of a]

parallel algorithm executed

on a parallel computer

SPEEDUP & EFFICIENCY

(Z)

Speedup = Serial exectime / RATIO

op⁵ performed by a parallel alg.
fall into 3 (ategories:

(i) serial computations

(z) parallel

inter-proc. com/tion, parallel overhead

using these 3 categories we can produce a simple model for speed-up.

W(np) speedup achieved solving a pb of size n or p proc.

o(n) inherently sequential (serial)
portion of the computation

6(n) portion of the computation that can be executed in para-lel

K(n,p) time required for parallel overhead.

Serial program on 1 proc. requires

time ocn + pcn + to execute the

(no inter-proc com/tion computations

required)

consider best possible parallel exections

time. Serial portion takes o(n) time (IDEAL)

parallel portion takes 6(n)/p time (IDEAL)

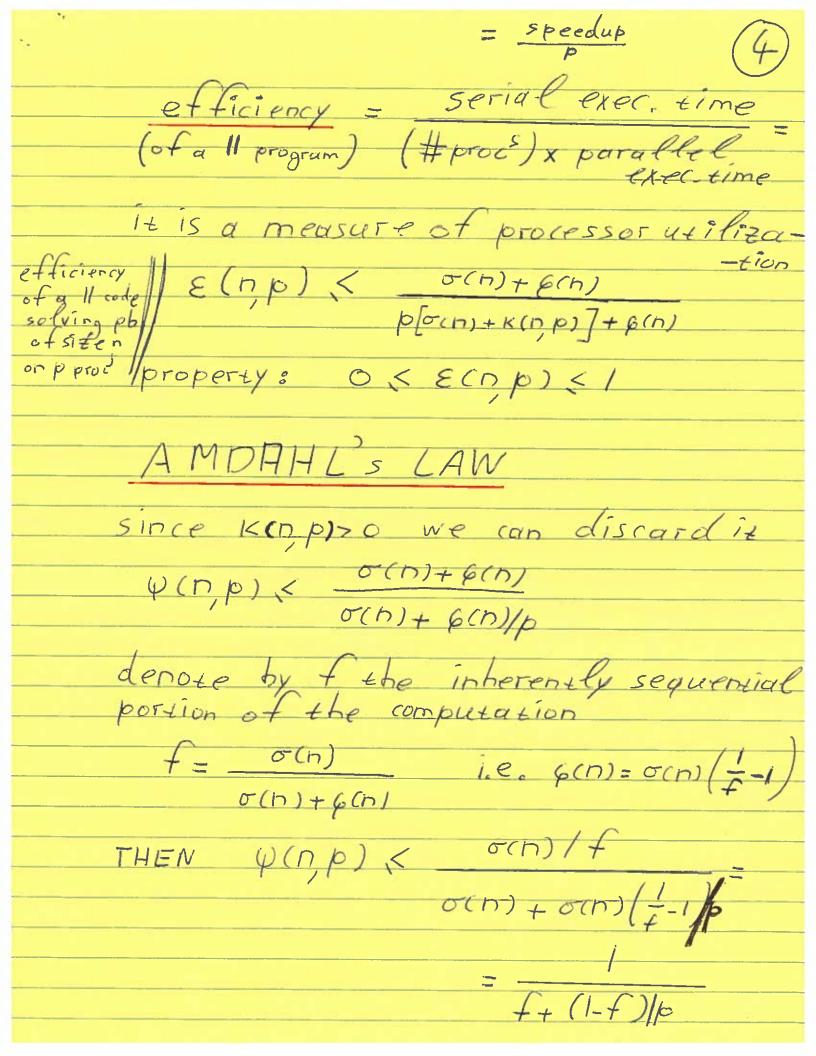
+ K(n,p) time for inter-proc. comm/tion ideal assumption: if parallelexectime
is larger, then speedup
is smaller. $\psi(n,p) < \frac{\sigma(n) + \rho(n)}{\sigma(n) + \frac{\rho(n)}{p} + \kappa(n,p)}$ general speed up formula adding procs reduces complian of parallel portion but increases com/tion at some point the comm/tion time increase is larger than the comp decrease so denom 1 ~> pcn/o)/

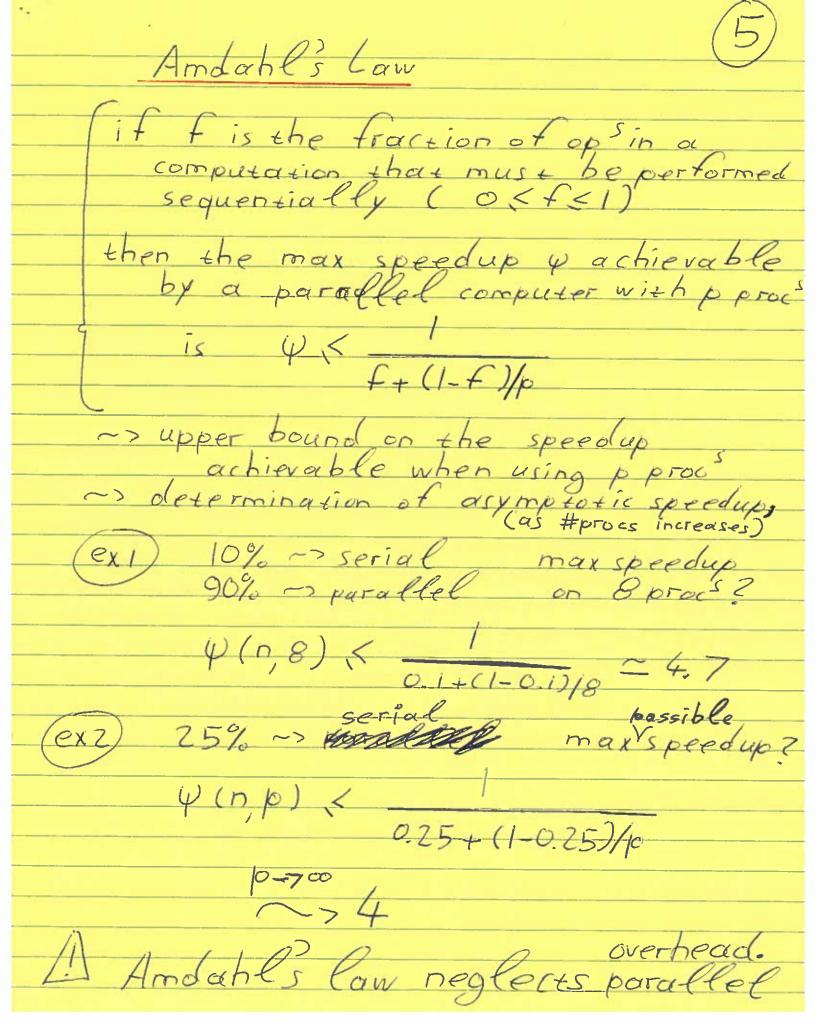
· Computation component decreoses

O COMMUNICATION COMPONENT INCFERSES

OLS # PROCE | FOR A FIXEP PB SIZE, THERE

IS AN OPTIMAL HORDES, MINIMIZING POPERTY.





Amdahl etteck for is O(gox) if Jro, MN sot. Vxxx0, N/gcm/</fr> 4 -> 9 bounds for above and below to gives tighter bounds on runtime than O-notation conseq. increasing size of pb,
increases the compation time
that than it increasses the
faster compatition time

Ama ~> for a fixed # procs Hmal speedup is usually an increasing fix of the pb. size. 4 6 8 10 12