# A parallel workload model and its implications for processor allocation


Allen B. Downey

University of California at Berkeley

San Diego Supercomputer Center

# Scheduling Parallel Jobs

Scheduling policy = queueing policy

+ allocation policy

+ preemption policy

- queueing: when do jobs run?
- allocation: how many processors?
- preemption: time-sharing?

# Allocation policy


- Rigid: fixed cluster size.
- Moldable: configurable at beginning, not later.
- Evolving: job-initiated changes in cluster size.
- Malleable: system-initiated changes in cluster size.

current workloads

# Conventional strategies

- AVG: allocate up to $A$ processors per job.
  - $A$ is average parallelism.
- A+: allocation decreases with load or $V$.
  - $V$ is variance in parallelism.
- PWS: choose cluster size to maximize power.
  - Power = speedup $\times$ efficiency.
- ASP: divide free processors among jobs in queue.
  - Maximum allocation of $A$.

# Conventional wisdom

- Maximum power (PWS) means optimal allocation.
  - Ghosal, Setia and Tripathi 93
- AVG is good because it approximates PWS.
  - Eager, Zahorjan and Lazowska 89
- Allocation should decrease linearly with load and *V* (A+).
  - Sevcik 89
- ASP minimizes average turnaround time under most workloads.
  - Chiang, Mansharamani and Vernon 94
- Memory limits impair adaptive scheduling.
  - Setia 95

# Outline

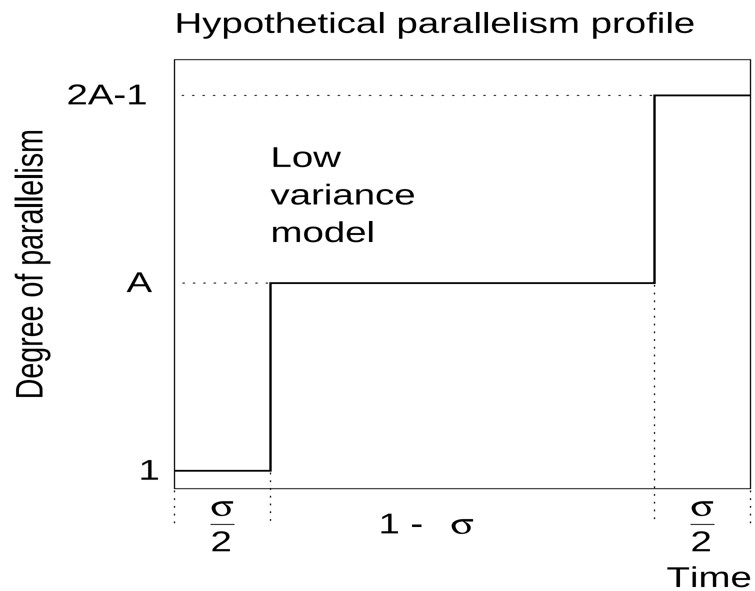
- Job model:  run time vs. cluster size
- Workload model:  distributions of parameters
- Simulation results

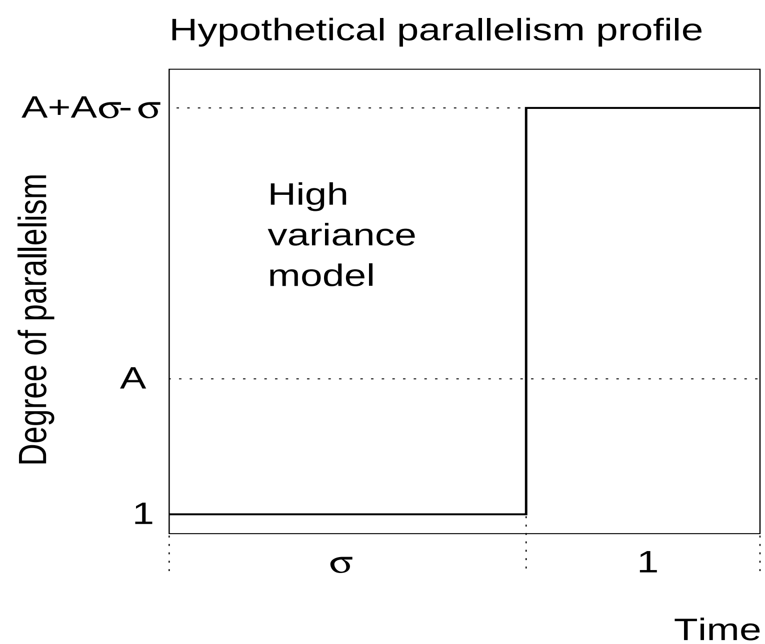
# Job Model

Program information:

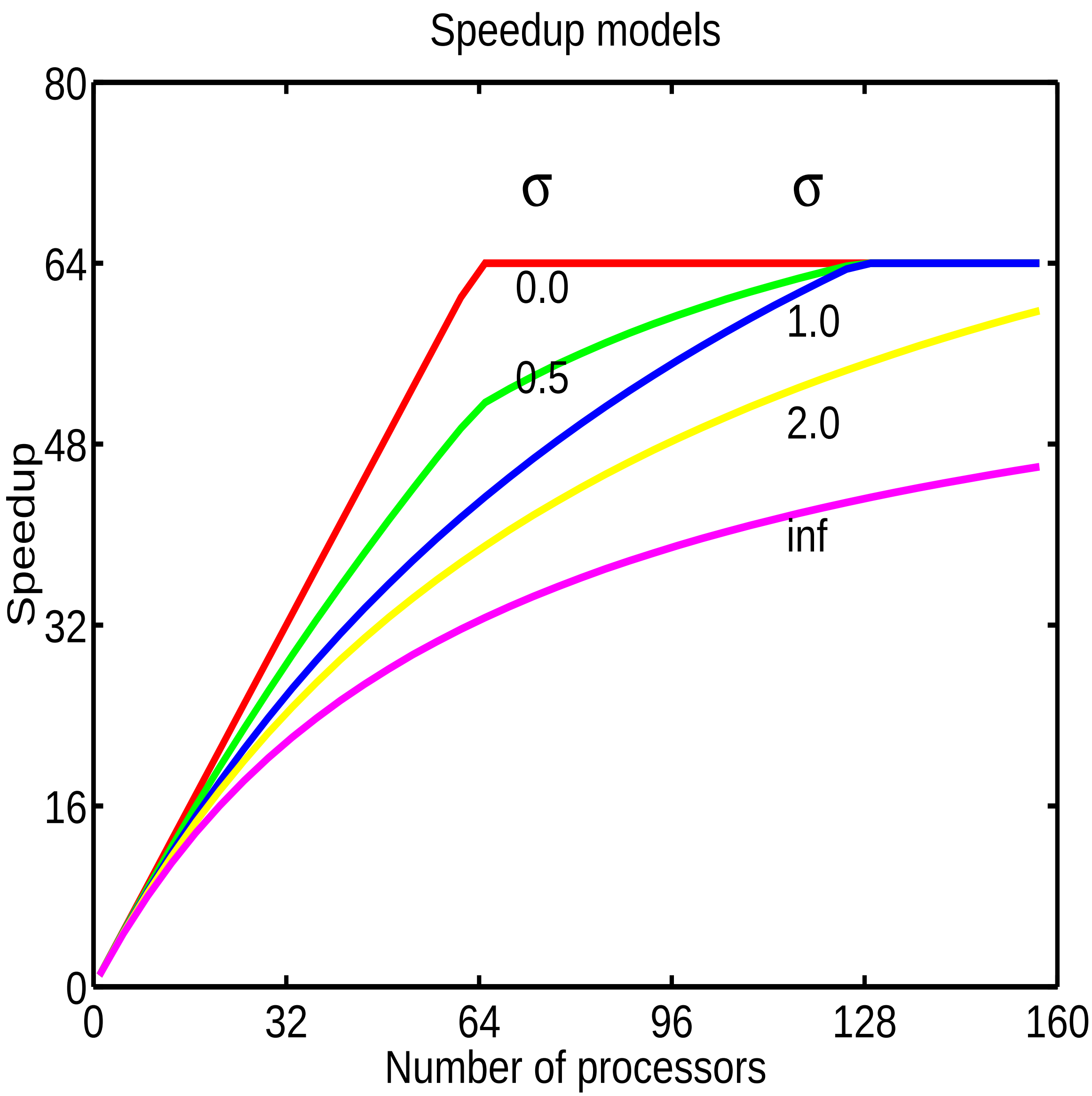
- Average parallelism $A$
- Variance in parallelism $V$
- Parallelism profile
- Speedup curve

For given $A$, $V$
assume profile.



# Speedup curves




- When σ=1, two models identical.
- For both models, $V = \sigma\,(A - 1)^2$
- $\sigma \approx CV^2$ of parallelism.
- Job parameters: $A$ and σ

# Abstract workload model

- Observed workloads
  - Distribution of lifetimes, cluster sizes, etc.
  - Depends on job mix, system, scheduler.

Wrong to use concrete workload from one system to evaluate another.

- Abstract workload model
  - Distributions of job parameters.
  - Separates job mix from system characteristics.

# Workload model

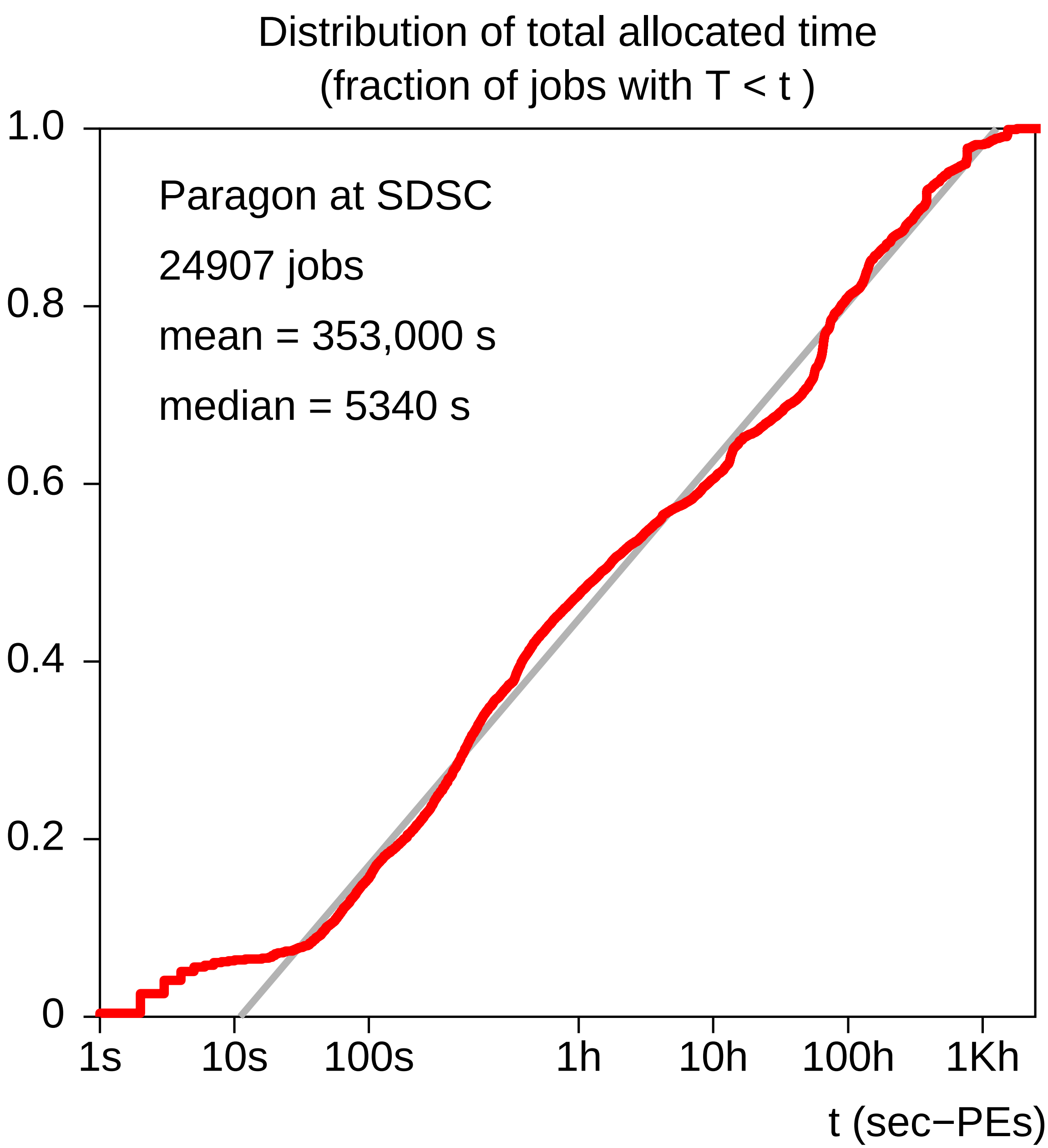
- Choose job parameters $L$, $A$, $\sigma$ from distributions.
  - $L$ is sequential lifetime.
  - Run time = $L / S(n, A, \sigma)$.

total allocated time ~ uniform-log → L uniform-log distribution

cluster sizes ~ discrete uniform-log → A uniform-log distribution

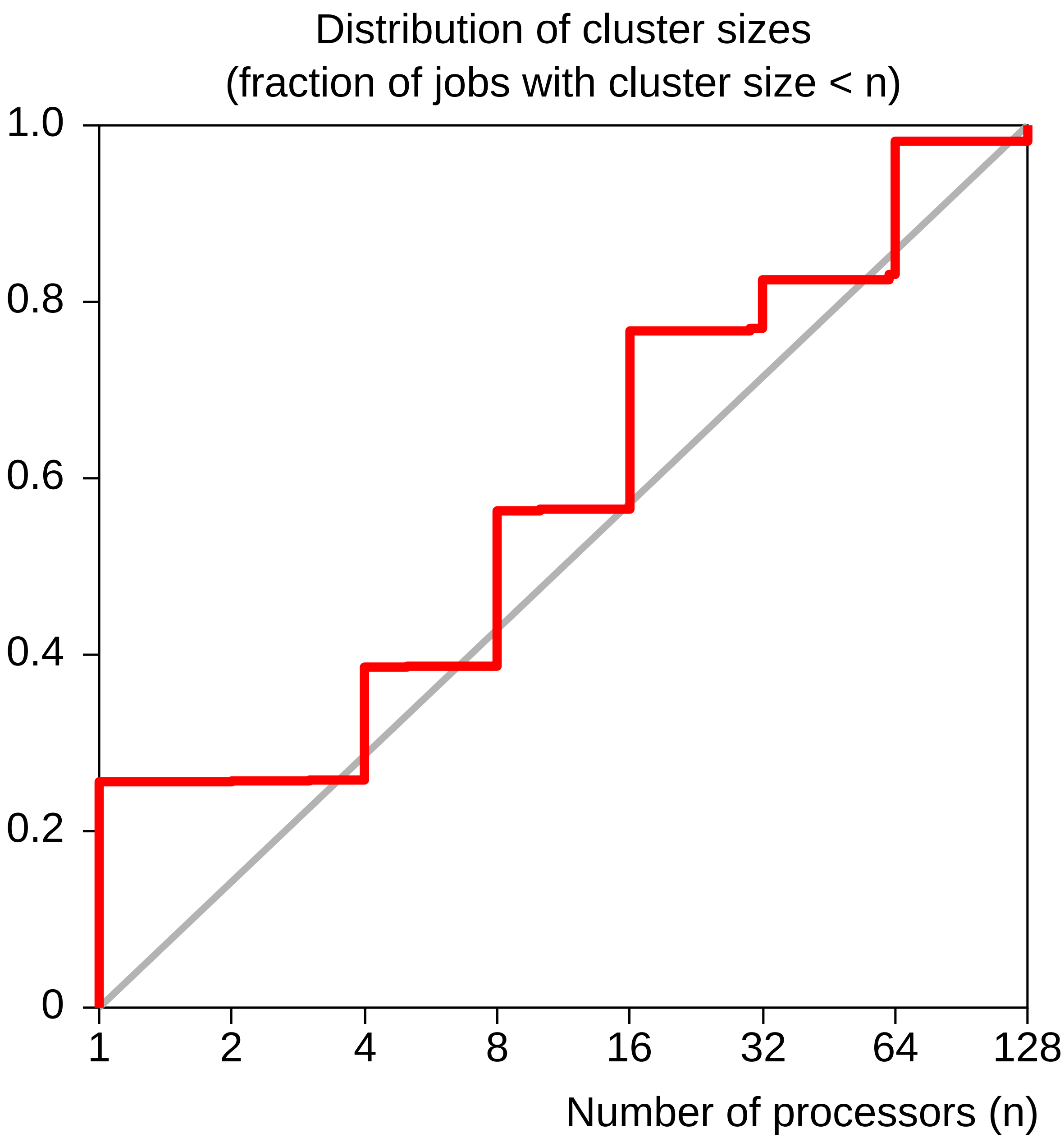
measured speedups $\sigma$ ~ [0, 2] → $\sigma$ uniform distribution

# Distribution of total allocated time



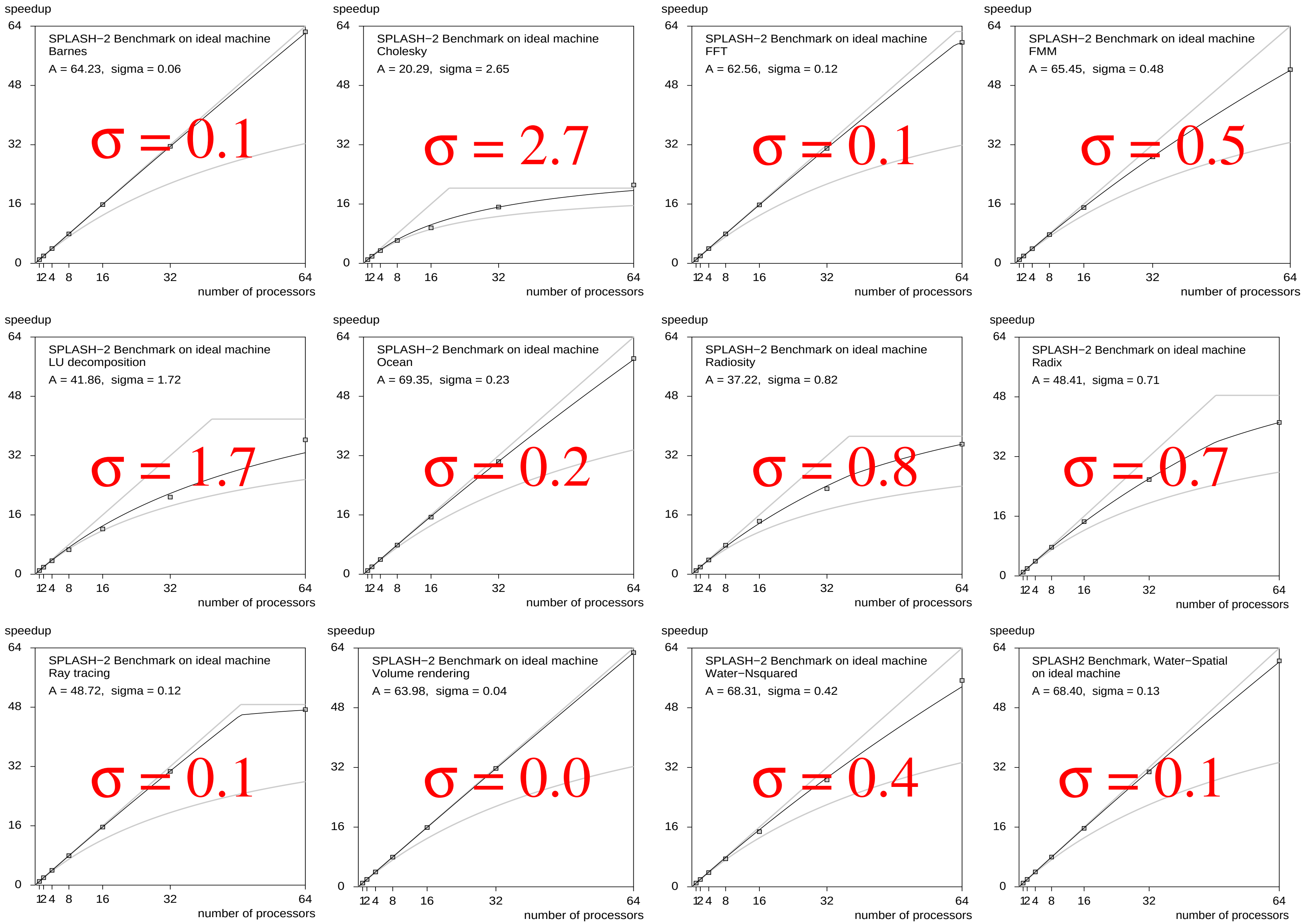

- Uniform on log axis.
- Same for SP2 at CTC.
- For non-linear speedup, $T > L$.
- Same shape distribution.

# Distribution of average parallelism




- Uniform on log axis, but discretized.
- Similar at CTC.
- Do cluster sizes reflect available parallelism?

# Distribution of σ



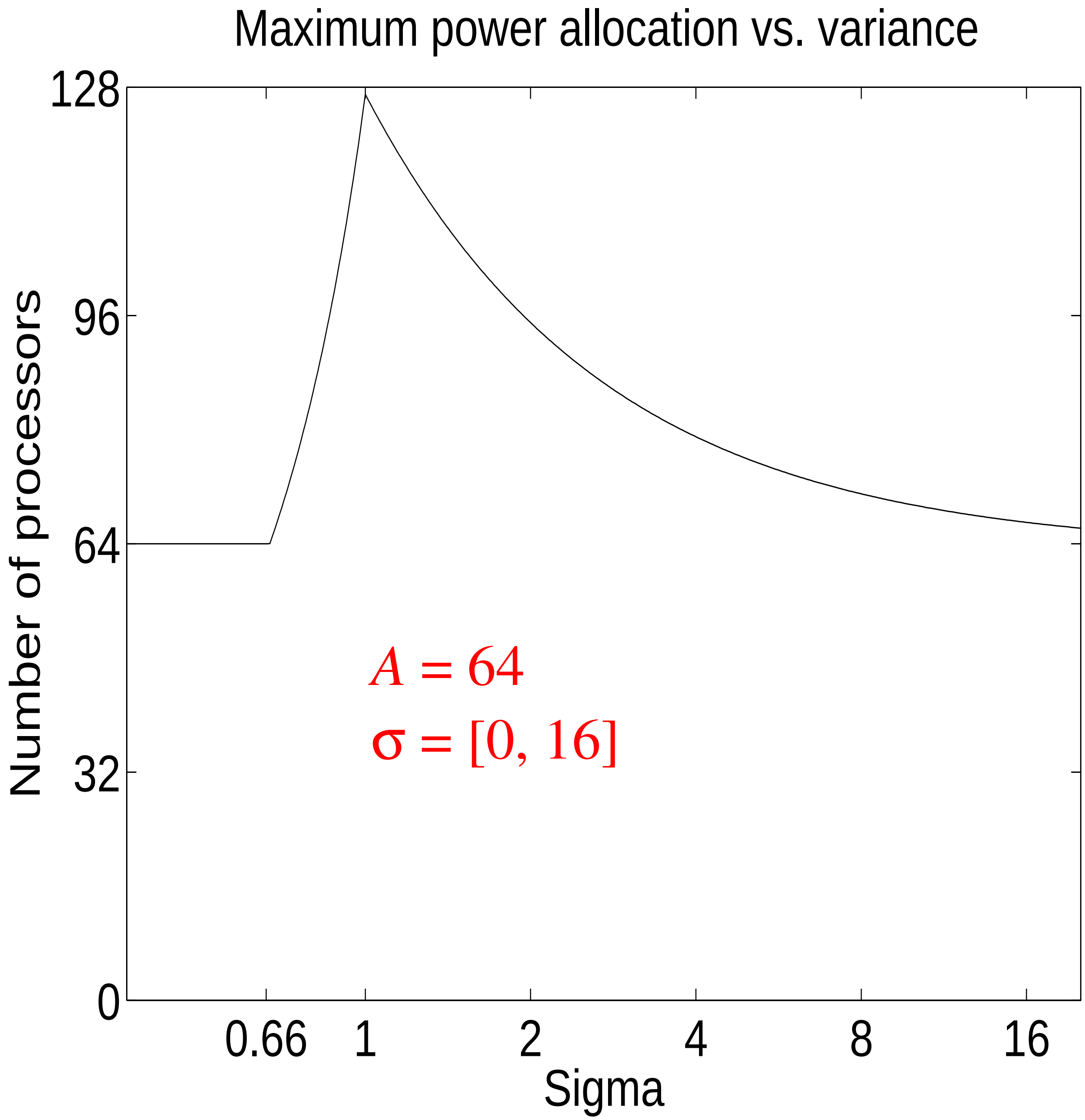
# Simulations

- Same workload for each strategy: 120 days, 80-150 jobs/day.
- Daily variation in arrival rate.

- Queueing policy: FIFO
- Allocation policies: AVG, PWS, A+, ASP
- Preemption policy: run-to-completion.

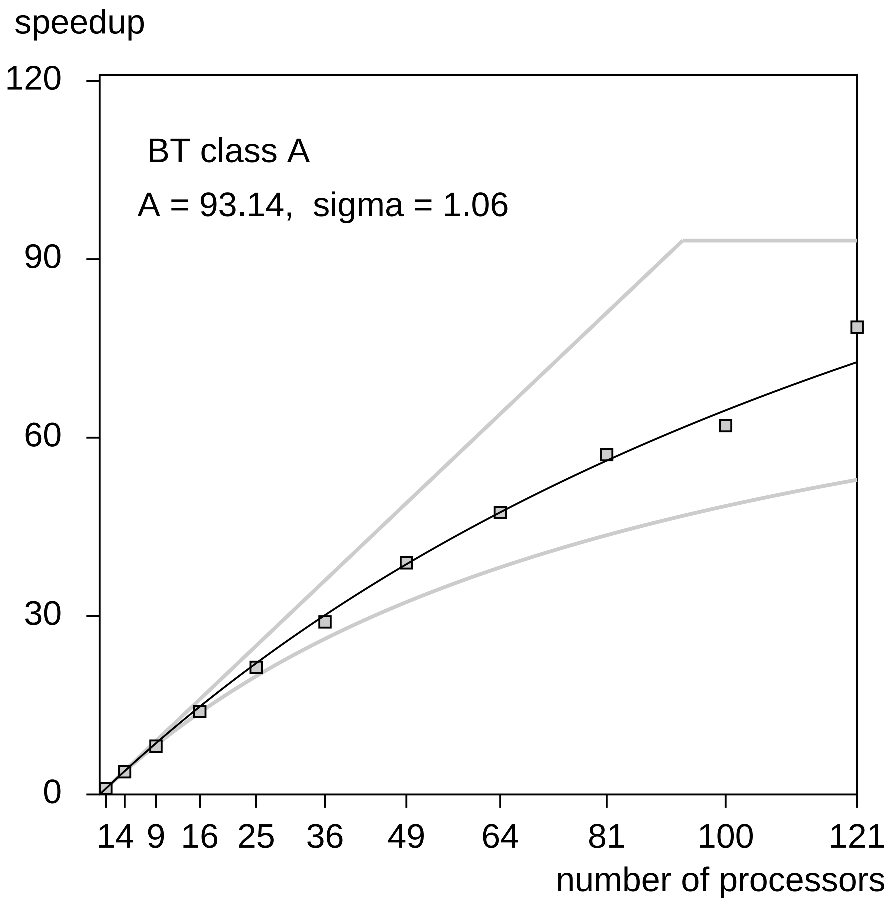
- Metrics:
  - Average utilization (over time and processors).
  - Average turnaround time (over jobs).
  - Average slowdown (over jobs).

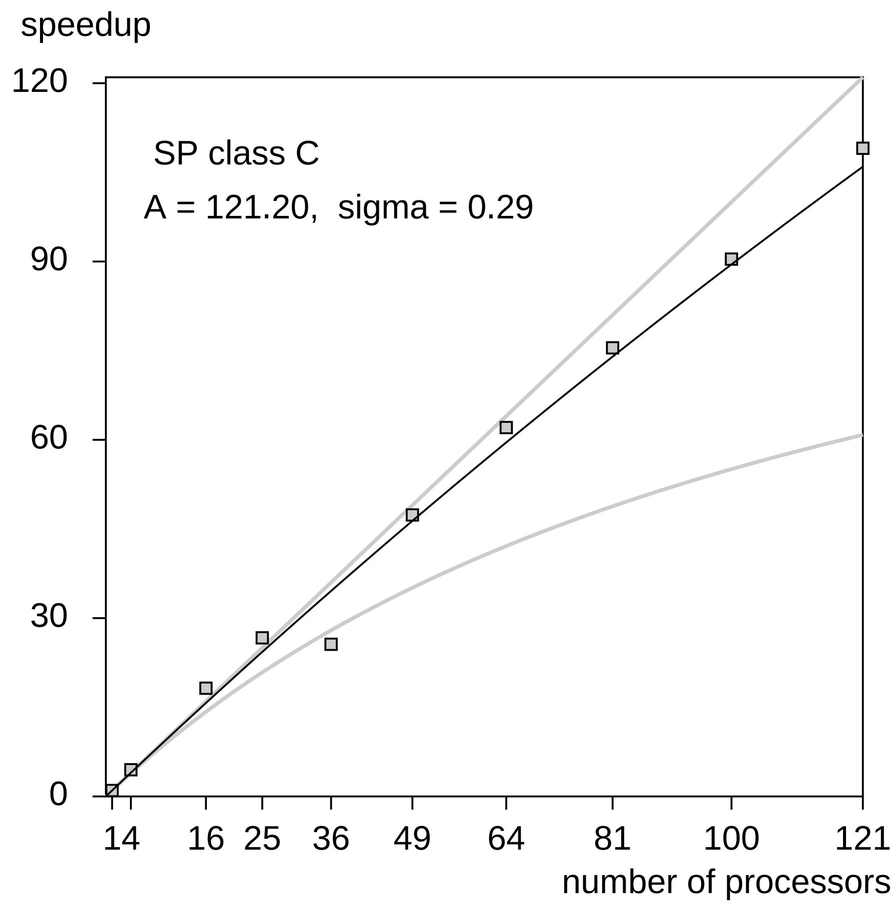
# Maximum power = optimal allocation?




- power = $S(n) \cdot e(n)$
- Small change in $\sigma$, big change in allocation.
- Violates $A+$ rule.

# Optimal allocation



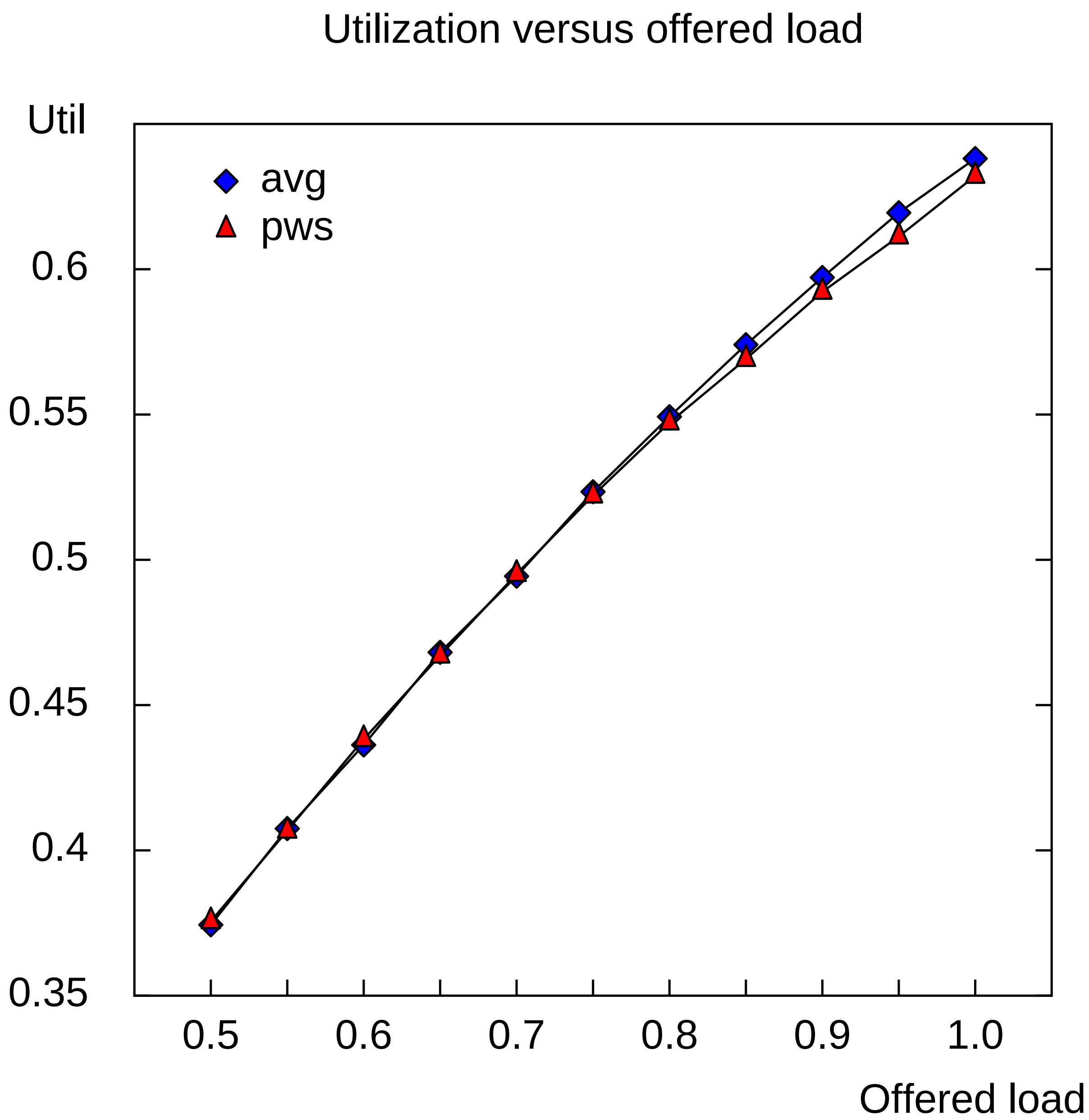

$A = 93, \sigma = 1.1$

pws = 177

$A = 121, \sigma = 0.3$

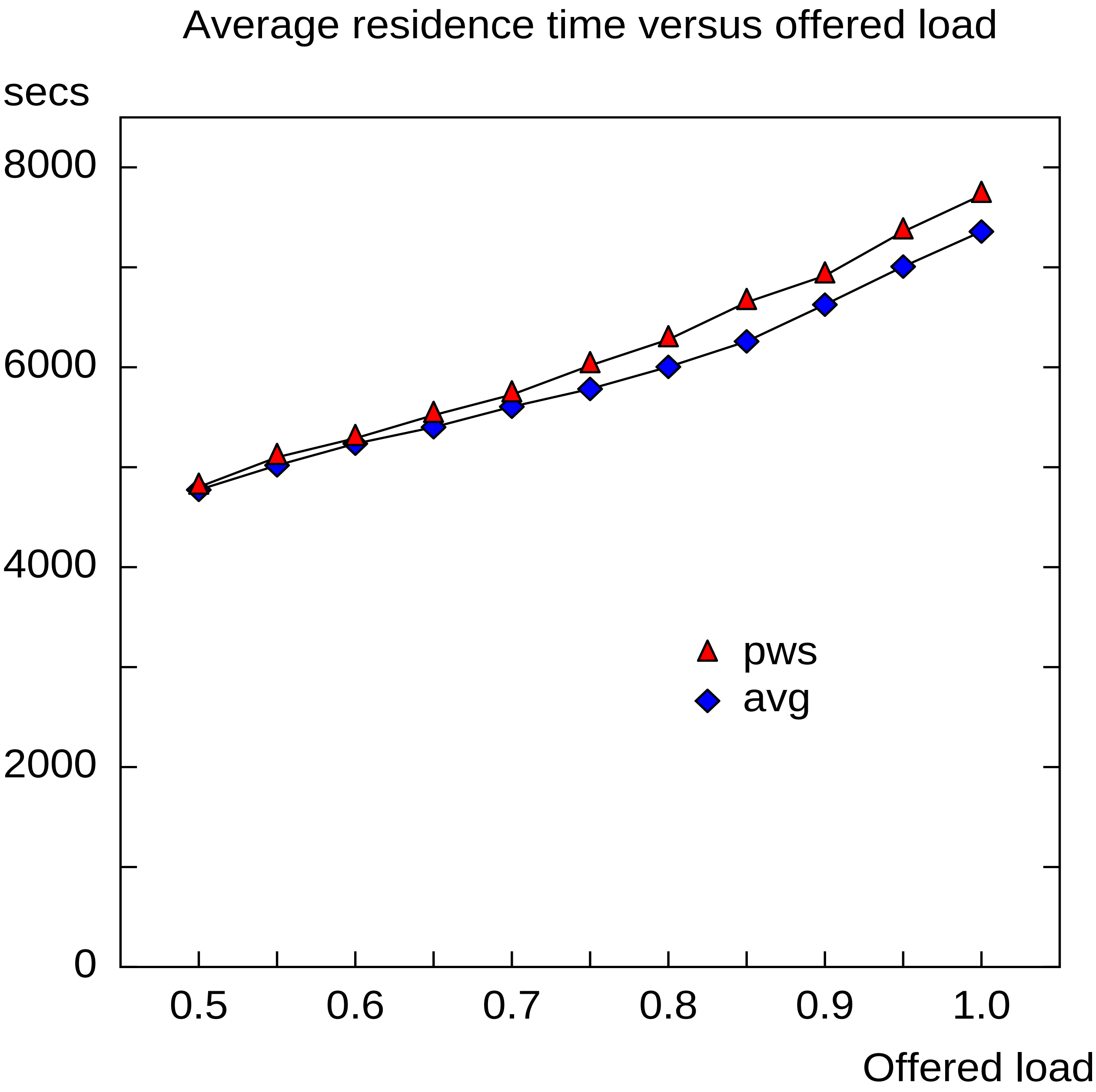
pws = 121

# Nope.




- AVG slightly better than PWS.
- Also true for larger systems, higher parallelism, different distributions of σ.

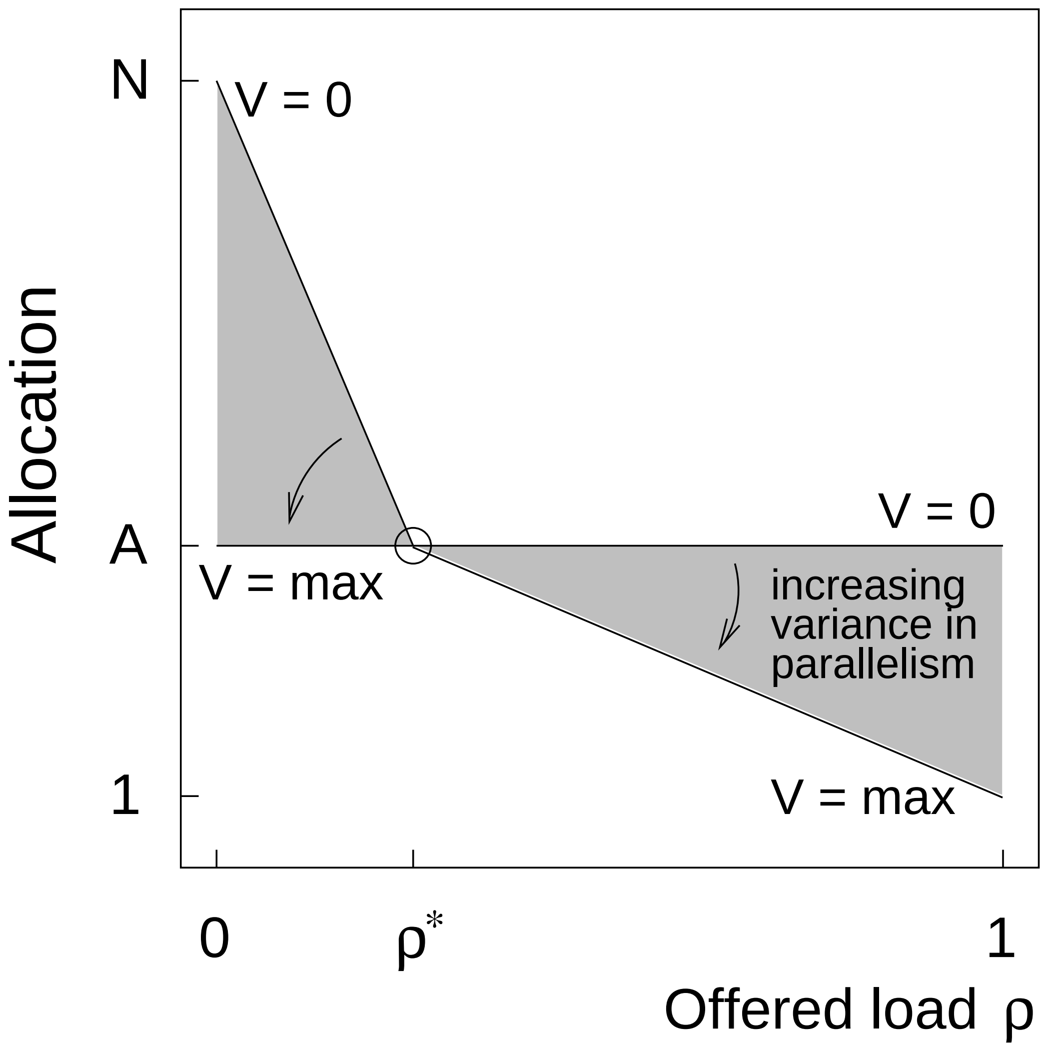
# Average turnaround time




- Average turnaround time and utilization do not conflict.

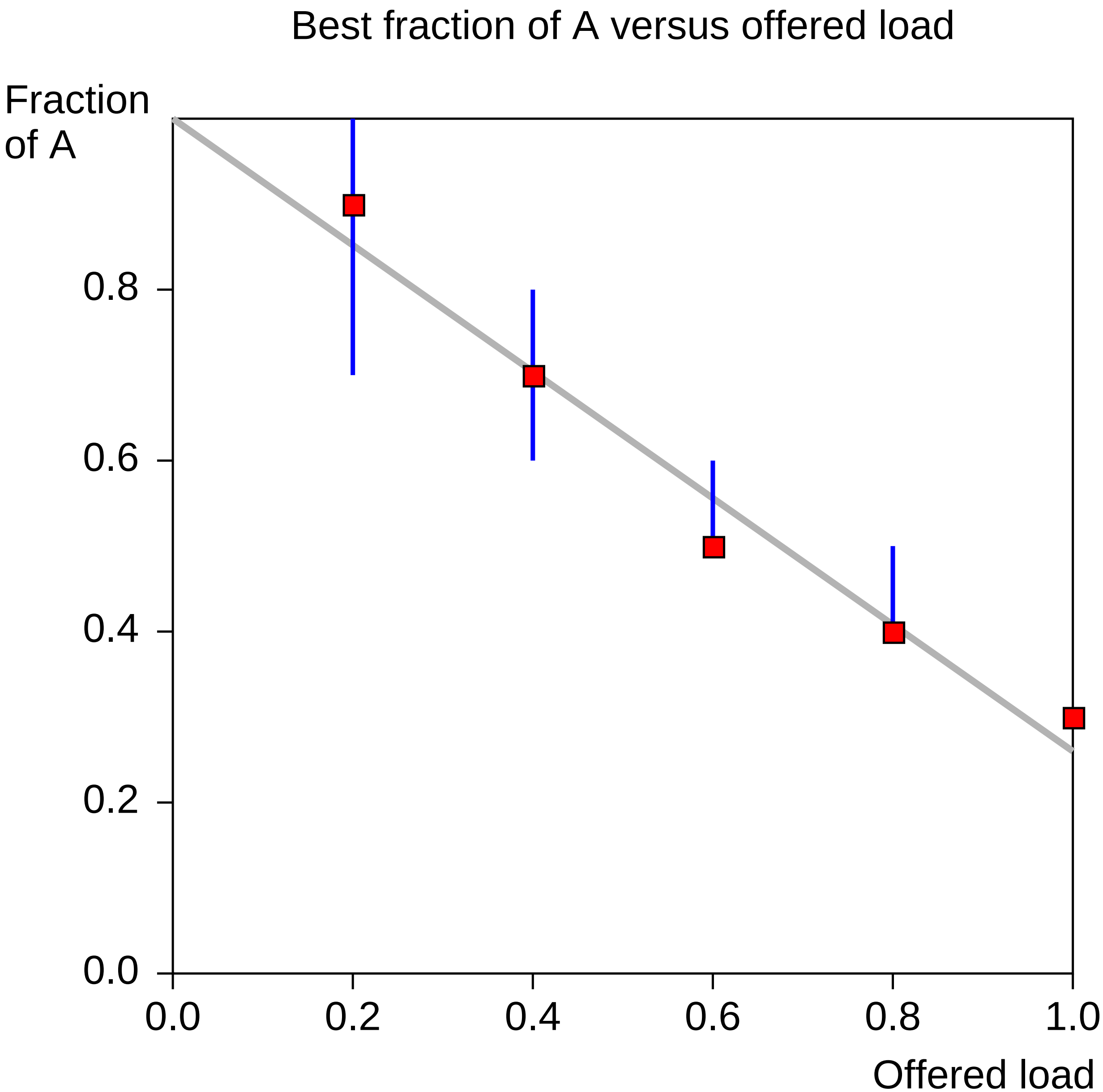
# Sevcik's method

Allocation as a function of load
and variance in parallelism




- For low loads, allocation varies from $A$ to $N$.
- For high loads, from 1 to $A$.
- $\rho^*$ is a free parameter, depends on workload.

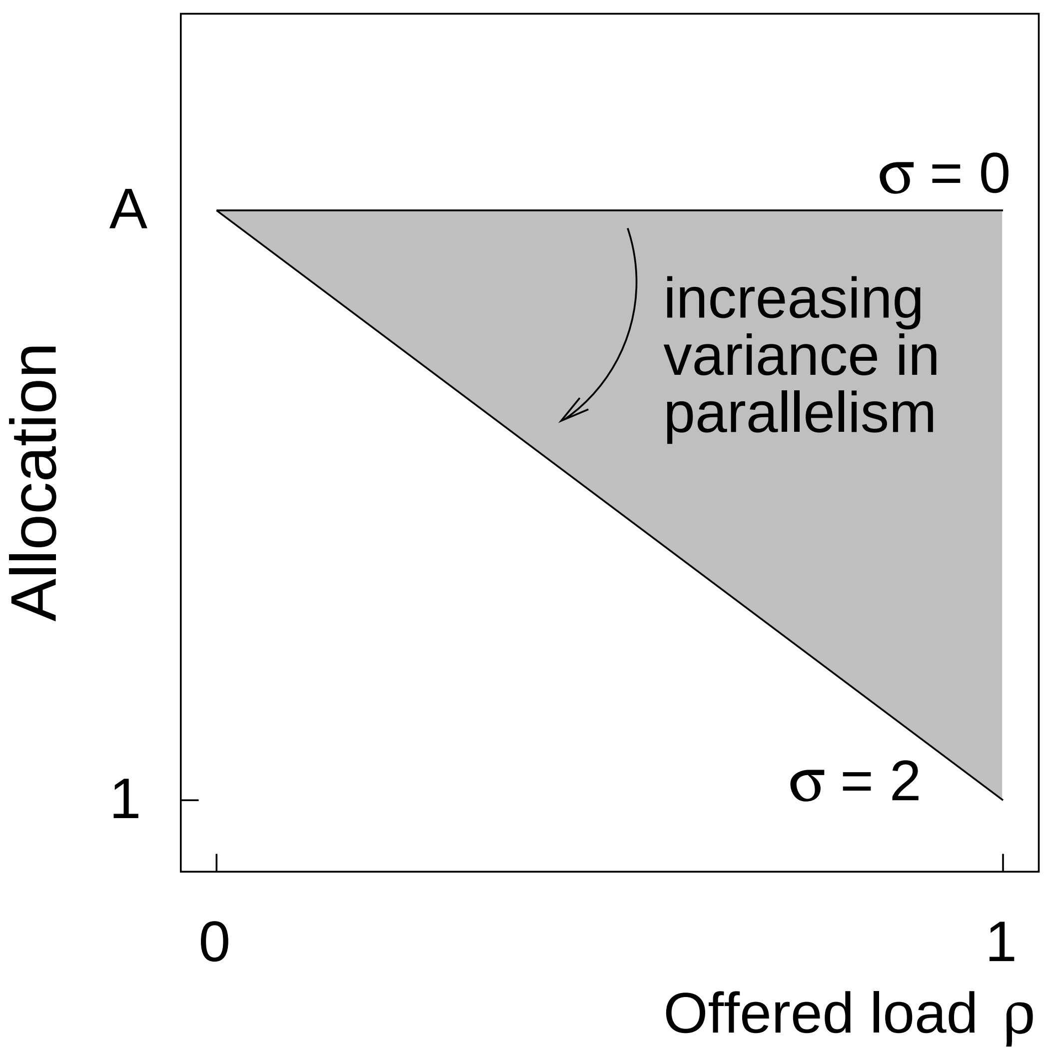
# Finding ρ*




- For various loads, find the fraction of *A* that maximizes utilization.
- ρ* is the load at which the best allocation is *A*.
- ρ* = 0

# Simplified Sevcik

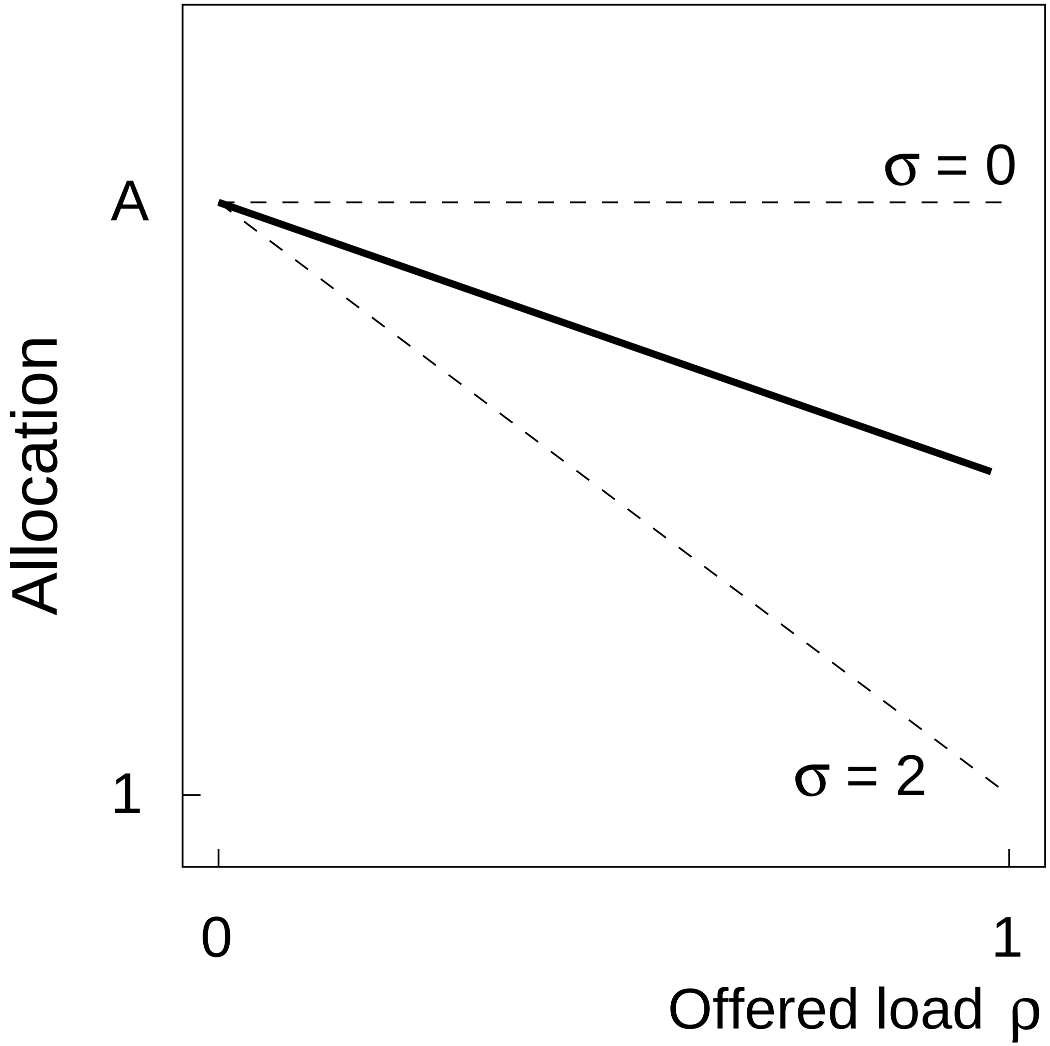
Allocation as a function of load
and variance in parallelism




- Allocation decreases linearly with load and $\sigma$.
- This version of A+ called SEV.

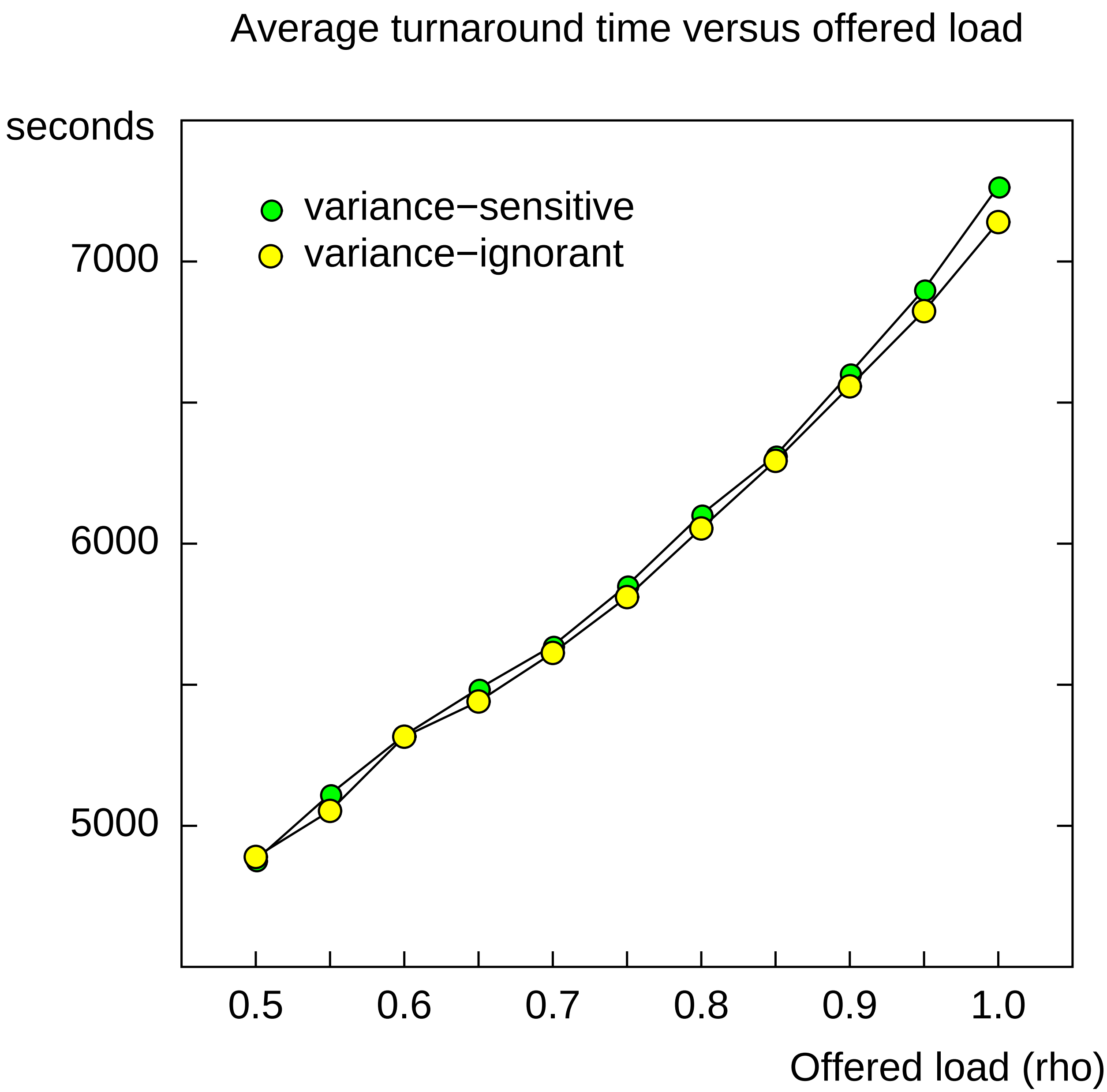
# Even simpler

Allocation as a function of load
and variance in parallelism



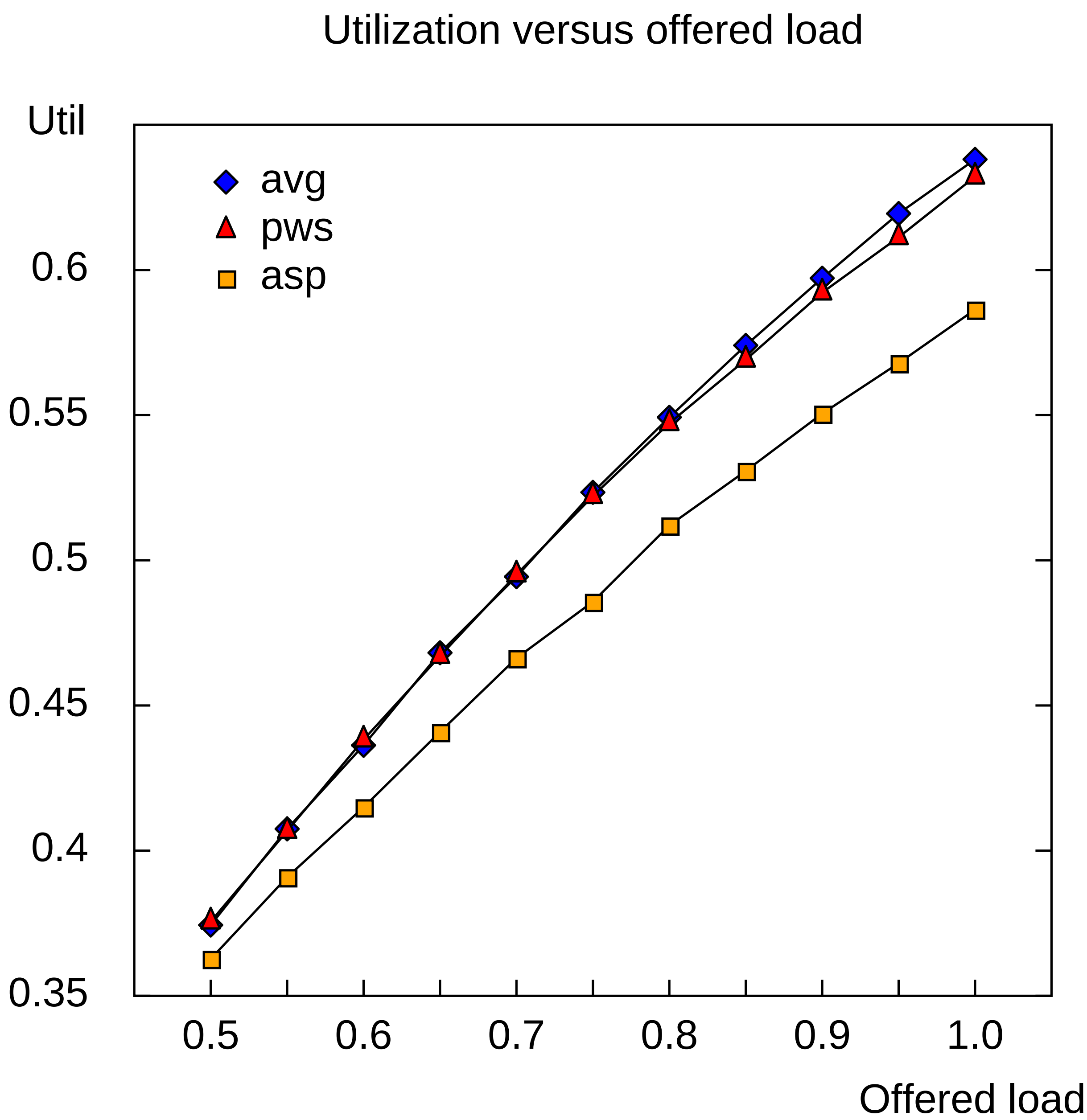

- Ignore variance.
- Same cluster sizes, on average.
- Same performance?

# Same performance



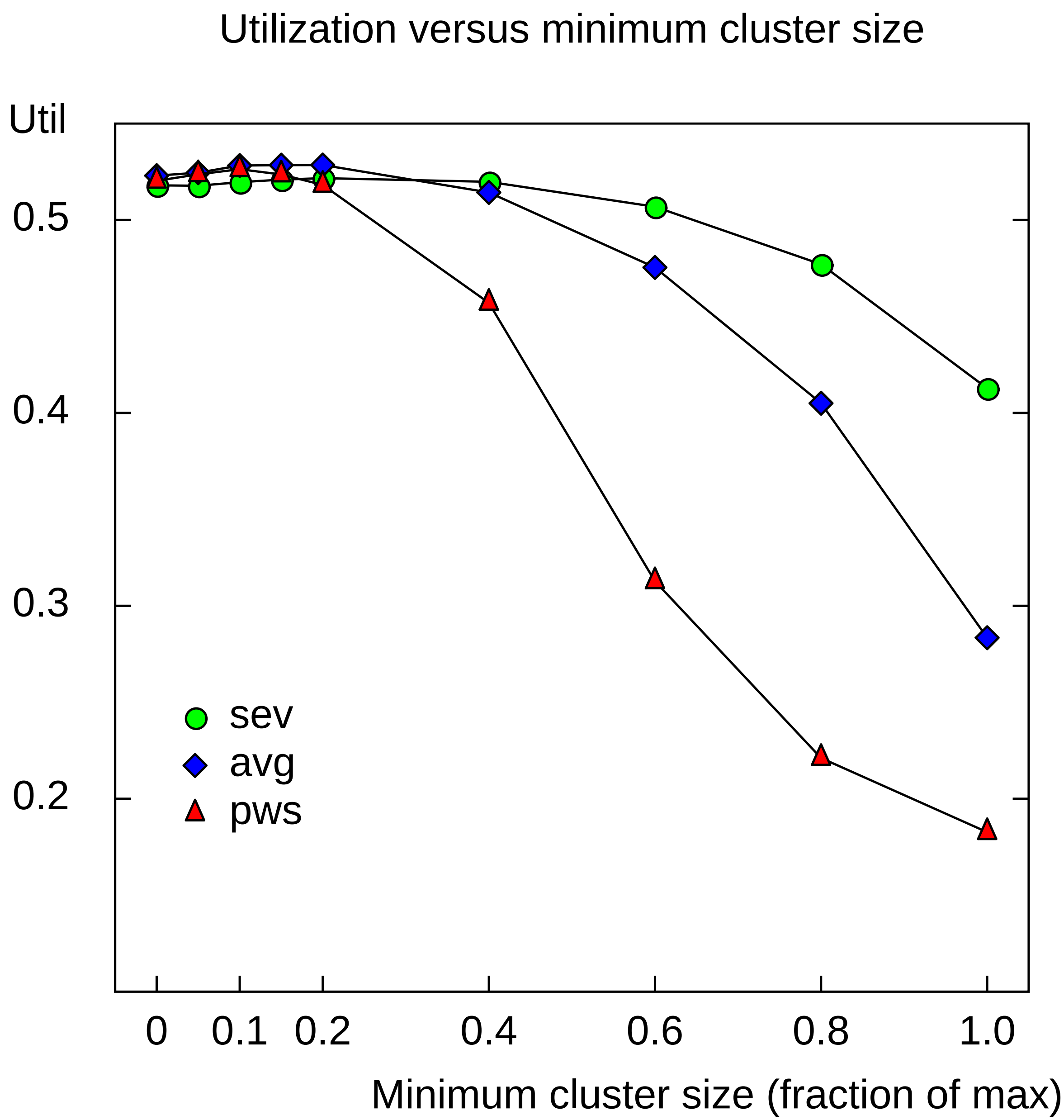

- Variance information yields slight, if any, advantage.

# How good is ASP?




- 30% longer turnaround time than AVG.
- Consistent with previous studies, when workload has high variance in parallelism.
- ASP overreacts to surges and lulls.

# Memory bounds




- If jobs require minimum cluster sizes, queue times increase and utilization decreases.
- Sevcik's method most tolerant of memory constraints.

# Other Observations

- Wide tradeoff between:
  - large clusters, short run times, long queue times.
  - small clusters, long run times, short queue times.

- Minimizing queue times minimizes slowdowns.
  - Until run times get ridiculous.
  - Or memory bounds take over.

$$\text{slowdown} = \frac{\text{actual run time}}{\text{shortest run time}}$$

# Conventional wisdom, review

- Maximum power (PWS) means optimal allocation.
  - False, no such thing as optimal allocation.
- AVG is good because it approximates PWS.
  - False, AVG is slightly better than PWS.
- Allocation should decrease linearly with load and *V* (A+).
  - Load, true. Variance, false.

# More conventional wisdom

- ASP minimizes average turnaround time for most workloads.
  - Maybe true, but not for this workload.

- Memory limits impair adaptive scheduling.
  - True, but some strategies more than others.