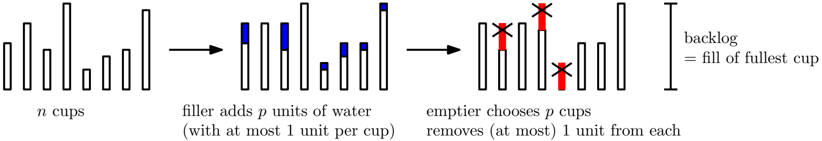
The Variable Processor Cup Game

Alek Westover

Belmont High School

June 7, 2020

p-PROCESSOR CUP GAME ON n CUPS



- ***filler*** adds water
- ***emptier*** removes water

Why is the Cup Game Important?

Models ***work scheduling***:

- Cups represent tasks
- At each time step:
  - new work comes in distributed among the tasks (*filler*)
  - must allocate processors to work on tasks (*emptier*)

Analysis of the cup game

Prove ***lower bounds***

- Exhibit a filling strategy that achieves large backlog (against any emptier)

Prove ***upper bounds***

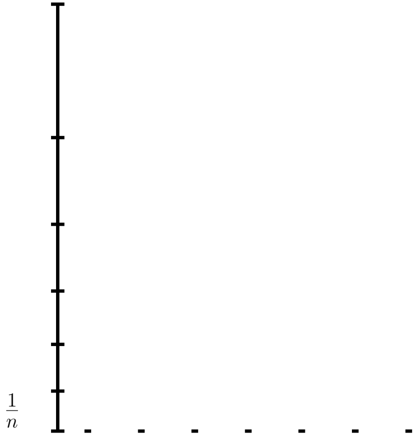
- Exhibit an emptying strategy that prevents backlog from growing large (against any filler)

Quick example: Single-processor cup game:

- $\Omega(\log n)$ lower bound
- $O(\log n)$ upper bound

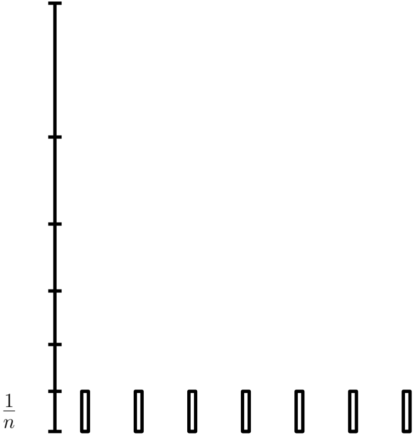
Single-Processor Lower Bound

Filling strategy:
Distribute water equally amongst cups not yet emptied from.



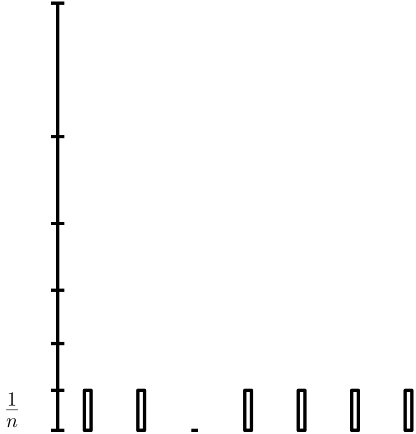
Single-Processor Lower Bound

Filling strategy:
Distribute water equally amongst cups not yet emptied from.



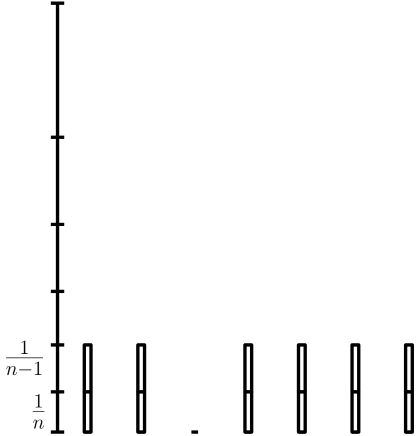
SINGLE-PROCESSOR LOWER BOUND

Filling strategy:
Distribute water equally amongst cups not yet emptied from.



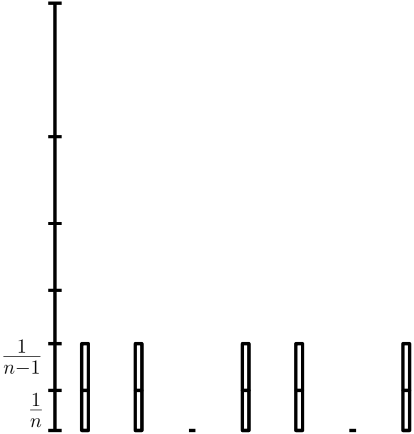
Single-Processor Lower Bound

Filling strategy:
Distribute water equally amongst cups not yet emptied from.



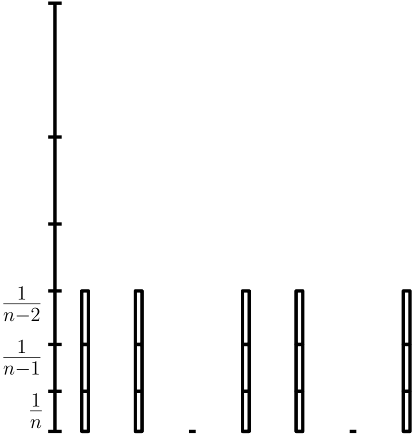
Single-Processor Lower Bound

Filling strategy:
Distribute water equally amongst cups not yet emptied from.



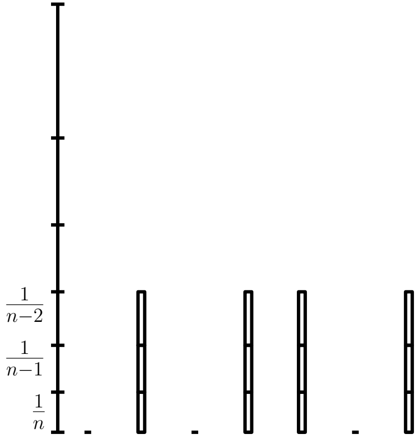
Single-Processor Lower Bound

Filling strategy:
Distribute water equally amongst cups not yet emptied from.



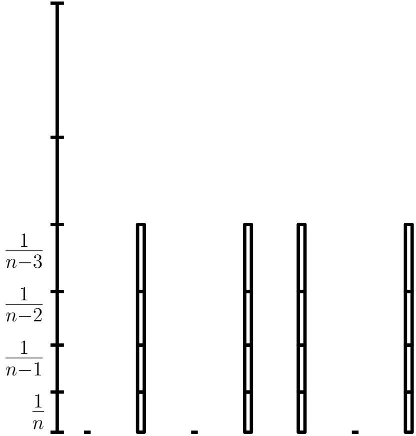
SINGLE-PROCESSOR LOWER BOUND

Filling strategy:
Distribute water equally amongst cups not yet emptied from.



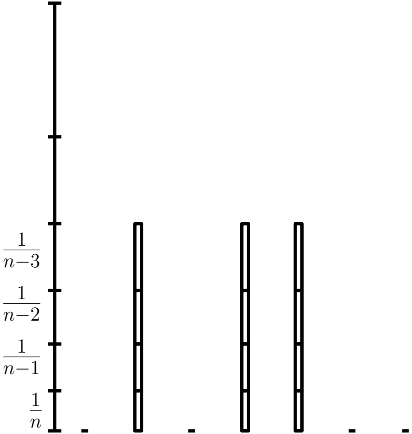
Single-Processor Lower Bound

Filling strategy:
Distribute water equally amongst cups not yet emptied from.



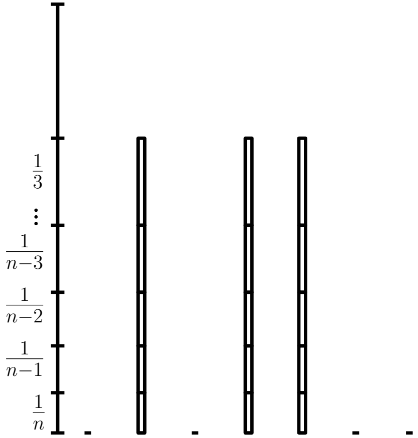
Single-Processor Lower Bound

Filling strategy:
Distribute water equally amongst cups not yet emptied from.



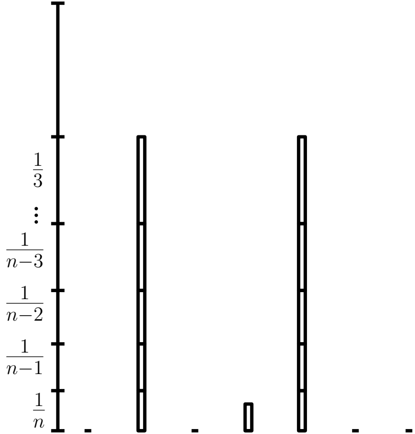
SINGLE-PROCESSOR LOWER BOUND

Filling strategy:
Distribute water equally amongst cups not yet emptied from.



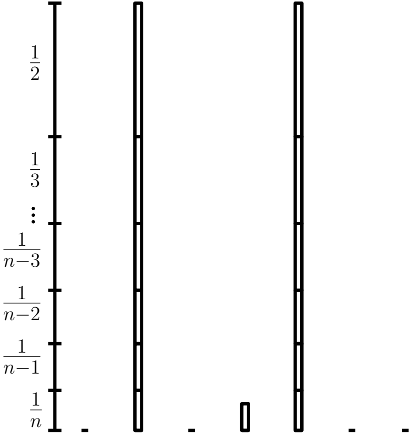
SINGLE-PROCESSOR LOWER BOUND

Filling strategy:
Distribute water equally amongst cups not yet emptied from.



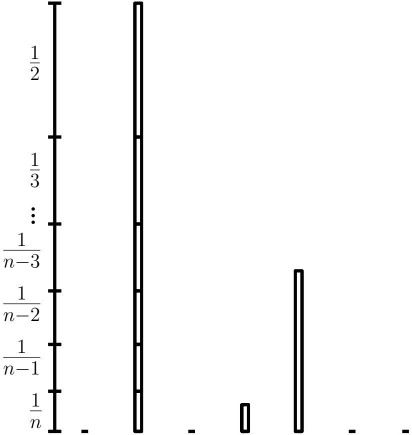
Single-Processor Lower Bound

Filling strategy:
Distribute water equally amongst cups not yet emptied from.



Single-Processor Lower Bound

Filling strategy:
Distribute water equally amongst cups not yet emptied from.



Single-Processor Lower Bound

Filling strategy:
Distribute water equally amongst cups not yet emptied from.

Achieves backlog:

$$\frac{1}{n} + \frac{1}{n-1} + \cdots + \frac{1}{2} = \Omega(\log n).$$

Single-Processor Upper Bound

A ***greedy emptier*** – an emptier that always empties from the fullest cup – never lets backlog exceed $O(\log n)$.

Definitions

- S_t: state at start of round t
- I_t: state after the filler adds water on round t, but before the emptier removes water
- $\mu_k(S)$: average fill of k fullest cups at state S.

Single-Processor Upper Bound Proof

Proof: Inductively prove a set of invariants:

$$\mu_k(S_t) \leq \frac{1}{k+1} + \ldots + \frac{1}{n}.$$

Let a be the cup that the emptier empties from on round t

If a is one of the k fullest cups in S_{t+1}:

$$\mu_k(S_{t+1}) \leq \mu_k(S_t).$$

Otherwise:

$$\mu_k(S_{t+1}) \leq \mu_{k+1}(I_t) \leq \mu_{k+1}(S_t) + \frac{1}{k+1}.$$

Our Variant

Definition

Variable-Processor Cup Game:
Each round filler can change p

Modification may seem small, but it drastically alters the game

Adaptive Filler Lower Bound

NEGATIVE FILL

In lower bound proofs we allow ***negative fill***

- Measure fill relative to average fill
- Important for recursion
- Strictly easier for the filler if cups can zero out

Amplification Lemma

Lemma

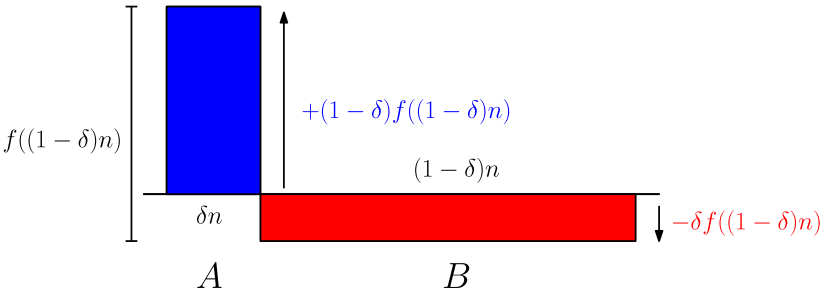
Given a strategy for achieving backlog $f(n)$ on n cups, we can construct a new strategy that achieves backlog

$$f'(n) \geq (1-\delta)\sum_{\ell=0}^{L} f((1-\delta)\delta^{\ell} n)$$

for appropriate parameters $L \in \mathbb{N}, 0 < \delta \ll 1/2$.

Amplification Lemma Proof Sketch

- A starts as the δn fullest cups, B as the $(1-\delta)n$ other cups.
- Repeatedly apply f to B and swap generated cup into A.
- Decrease p, recurse on A.

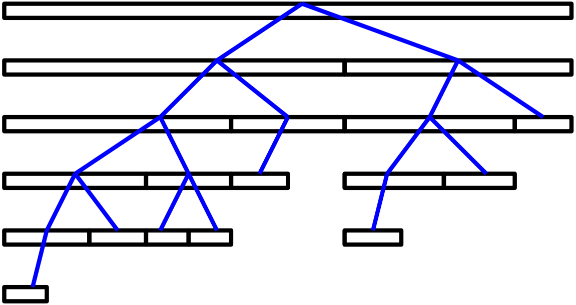


Adaptive Filler Lower Bound

Let $\epsilon > 0$ be any constant. Then there is some $\delta = \Theta(1)$ such that by repeated amplification we get:

Theorem

There is an adaptive filling strategy that achieves backlog $\Omega(n^{1-\epsilon})$ in running-time $2^{O(\log^2 n)}$.

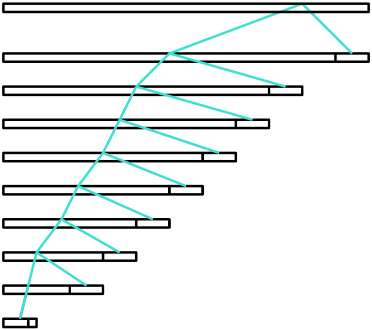


Adaptive Filler Lower Bound

Extremal strategy:
By repeated amplification using $\delta = \Theta(1/n)$ we get:

Theorem

There is an adaptive filling strategy that achieves backlog $\Omega(n)$ in running-time $2^{O(n)}$.



Upper Bound

Upper Bound

We prove a novel set of invariants:

Theorem

A greedy emptier maintains the invariant:

$$\mu_k(S_t) \leq 2n - k.$$

Corollary

A greedy emptier never lets backlog exceed

$$O(n).$$

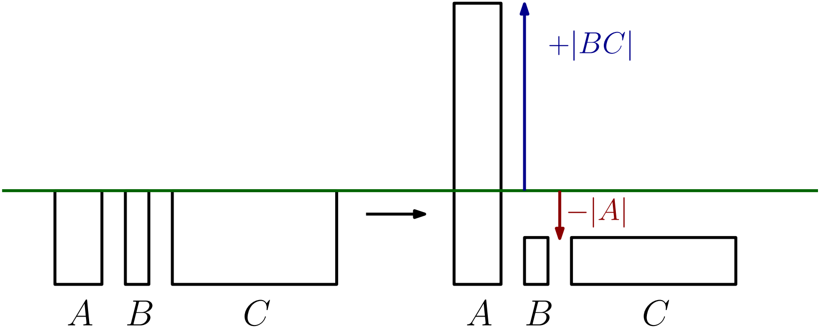
Note: this matches our lower bound!

Upper Bound Proof Sketch

Induct on t. Fix k. Define sets of cups:

- A: (emptied from) $\cap$ (k fullest in S_t) $\cap$ (k fullest in S_{t+1})
- B: (emptied from) $\cap$ (k fullest in S_t) $\cap$ (**not** k fullest in S_{t+1})
- C: AC is the k fullest cups in S_{t+1}

$\mu_k(S_{t+1})$ is largest if fill from BC is pushed into A



Oblivious Filler
Lower Bound

Oblivious Filler Lower Bound

Definition

Oblivious Filler: Can't observe the emptier's actions

- Classically emptier does better in the randomized setting.
- But not in the variable-processor cup game!
- We get the same lower bound as with an adaptive filler in quasi-polynomial length games!

Oblivious Filler Lower Bound

Definition

Δ-***greedy-like*** emptier:
Let x, y be cups. If $\text{fill}(x) > \text{fill}(y) + \Delta$ then a Δ-greedy-like emptier empties from y *only if* it also empties from x.

Oblivious filler can achieve backlog $\Omega(n^{1-\epsilon})$ for $\epsilon > 0$ constant in running time $2^{\text{polylog}(n)}$ against a Δ-greedy-like emptier ($\Delta \leq O(1)$) with probability at least $1 - 2^{-\text{polylog}(n)}$.

Flattening

Definition

A cup configuration is R-flat if all cups have fills in $[-R, R]$.

Proposition

Oblivious filler can get a $2(2+\Delta)$-flat configuration from an R-flat configuration against a Δ-greedy-like emptier in running time $O(R)$.

Oblivious Filler: Constant Fill

Getting constant fill in a *known* cup is hard now. Strategy:

- Play many single-processor cup games on $\Theta(1)$ cups blindly. Each succeeds with constant probability.
- By a Chernoff Bound with probability $1 - 2^{-\Omega(n)}$ at least a constant fraction nc of these succeed.
- Set $p = nc$.
- Fill nc known cups; because emptier is greedy-like it must focus on the nc cups with high fill before these cups.
- Recurse on the nc known cups with high fill.

Oblivious Amplification Lemma

Almost identical to the Adaptive Amplification Lemma!

Lemma

Given a strategy f for achieving backlog $f(n)$ on n cups, we can construct a new strategy that achieves backlog

$$f'(n) \geq \phi \cdot (1-\delta) \sum_{\ell=0}^{L} f((1-\delta)\delta^{\ell} n)$$

for appropriate parameters $L \in \mathbb{N}, 0 < \delta \ll 1/2$ and constant $\phi \in (0,1)$ of our choice.

(Note: Lemma is actually more complicated than this.)

Oblivious Filler Lower Bound

Theorem

There is an oblivious filling strategy that achieves backlog

$$\Omega(n^{1-\epsilon})$$

for constant $\epsilon > 0$ *with probability at least* $1 - 2^{-\operatorname{polylog}(n)}$ *in running time* $2^{O(\log^2 n)}$.

Achieve this probability by a union bound on $2^{\operatorname{polylog}(n)}$ events.

Open Questions

- Can we extend the oblivious lower bound construction to work with arbitrary emptiers?
- Are there shorter more simple constructions?

Acknowledgements

- My mentor William Kuszmaul
- MIT PRIMES
- My Parents

Question Slides