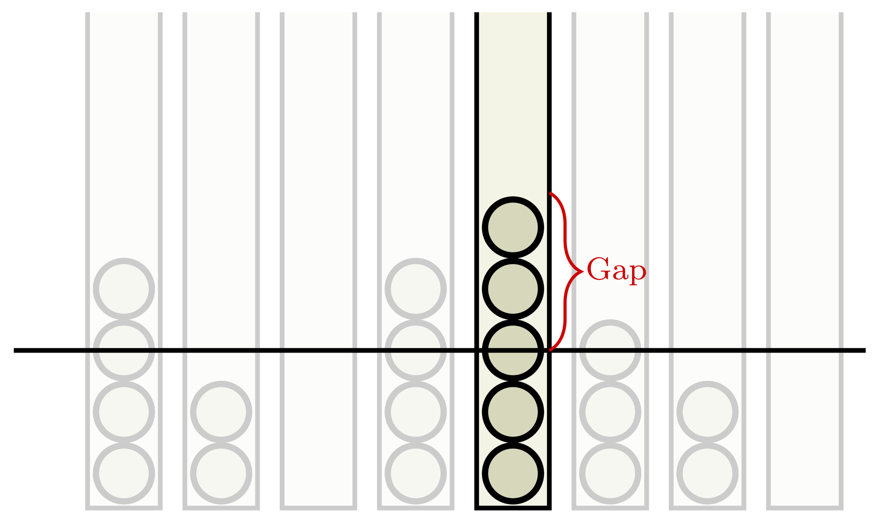
Balanced Allocations: Relaxing Two-Choice

Dimitrios Los (Cambridge), Thomas Sauerwald (Cambridge), John Sylvester (Glasgow)

Problem Formulation

In the *balanced allocations setting*,

- Task: Allocate m tasks (**balls**) sequentially into n machines (**bins**).
- Goal: Minimize the **maximum load** $\max_{i \in [n]} x_i^m$, where x^t is the load vector after ball t.



centralized setting → use round-robin
decentralized setting → use randomized approaches

Power of Two Choices

One-Choice: each ball is allocated in a bin sampled uniformly at random.

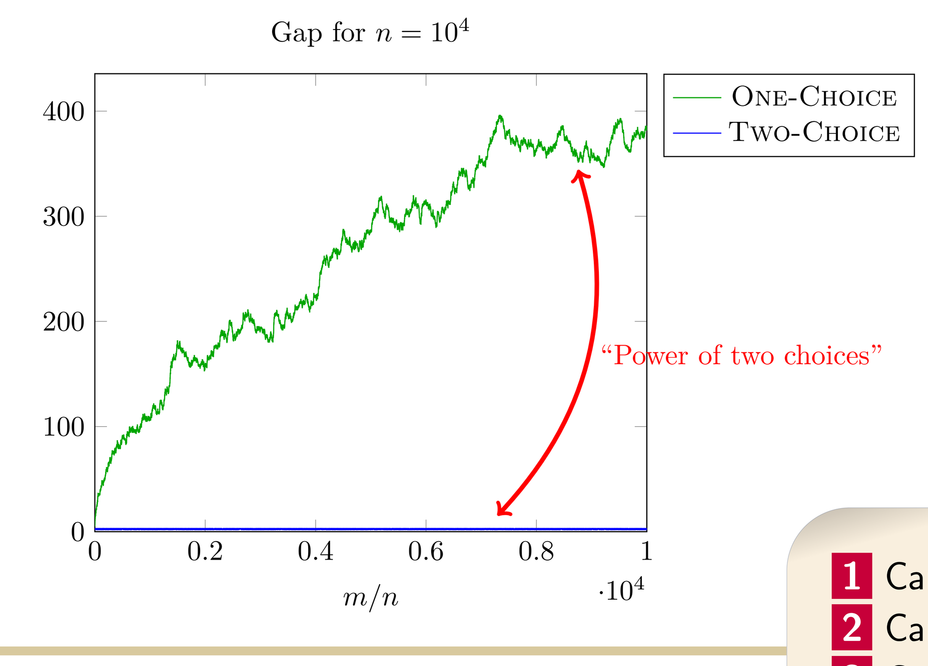
- For any $m \geq n \log n$ (e.g., [9]):

$$\text{Gap}(m) = \Theta\left(\sqrt{\frac{m}{n} \cdot \log n}\right).$$

Two-Choice: each ball is allocated in the lesser loaded of *two* bins sampled uniformly at random.

- For any $m \geq n$ ([1, 2, 3]):

$$\text{Gap}(m) = \log_2 \log n + \Theta(1).$$



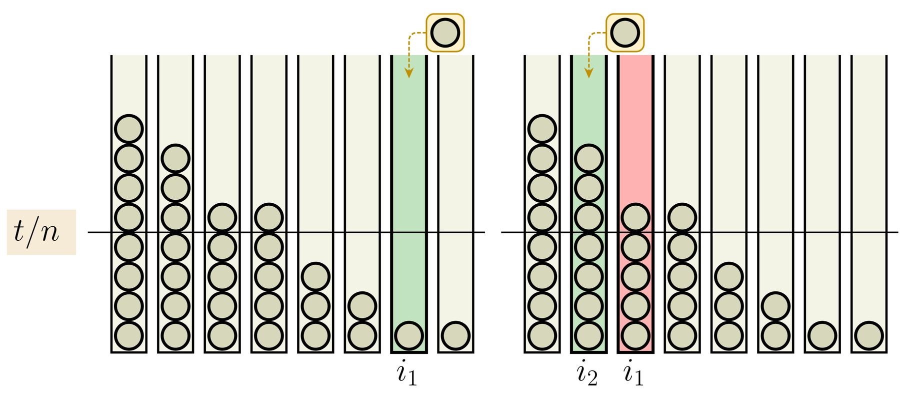
1 Can we use fewer than 2 *samples*?
2 Can we relax *synchronization*?
3 Can load values be *outdated*?

Processes

Mean-Thinning (1, 2)

For each ball:

- Sample one bin; if load at most t/n, then allocate.
- Otherwise, sample a second bin and allocate there.

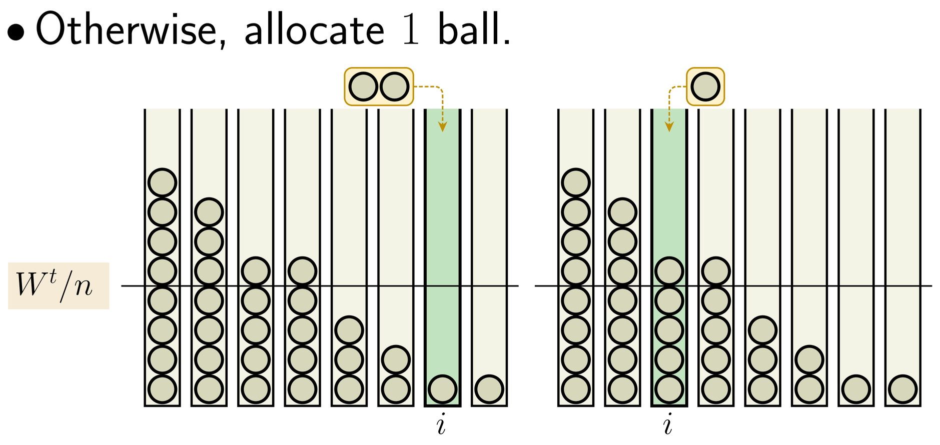


$\text{Gap}(m) = \Theta(\log n)$ [7] #Samples $= 2 - \epsilon$

Twinning (1, 2)

For each ball:

- Sample one bin; if load at most W^t/n, then allocate 2 balls.
- Otherwise, allocate 1 ball.

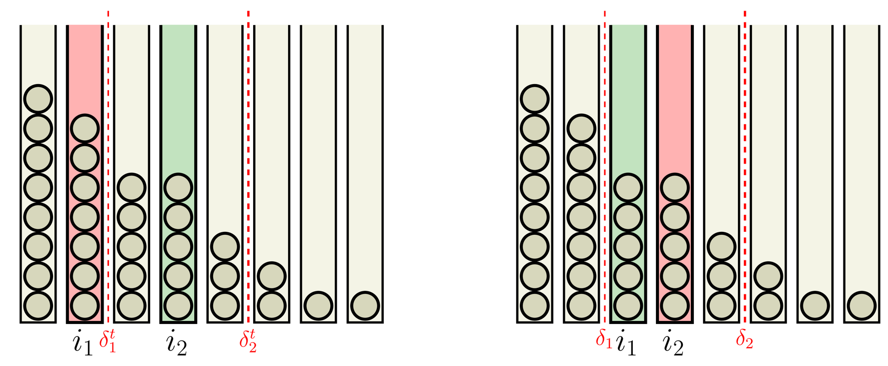


$\text{Gap}(m) = \Theta(\log n)$ #Samples $= 1 - \epsilon$

Quantile (2)

For each ball:

- Sample two bins i_1 and i_2.
- Send k queries of the form: *is load at median?*
- Based on responses, allocate to smaller one, or randomly.



$\text{Gap}(m) = \Theta(k \cdot (\log n)^{1/k})$ #Samples $= 2$

Packing: Extends this to add more than 2 balls to underloaded bins.

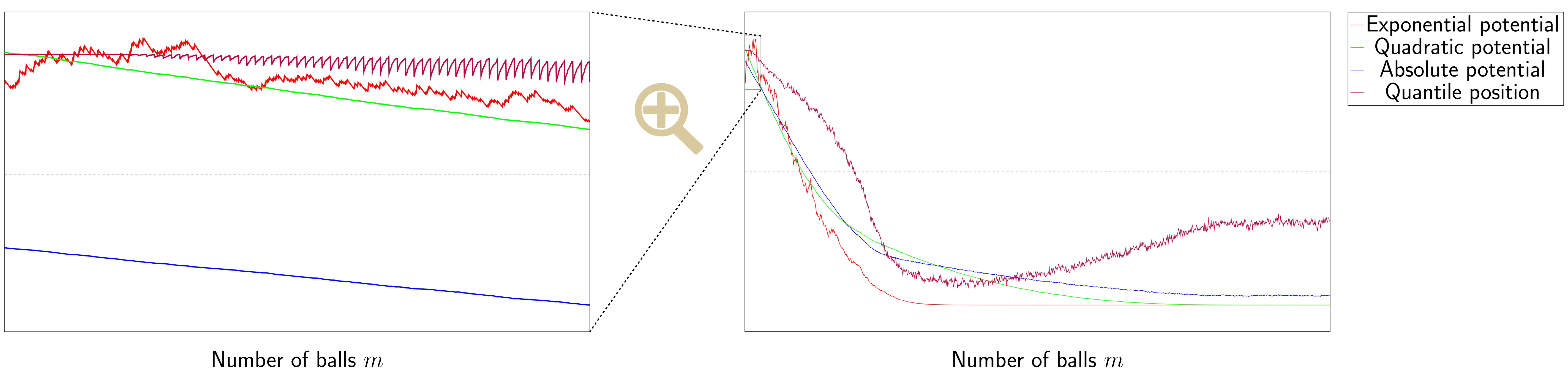
Outline of the Analysis for Mean-Thinning

Our analysis is based on an interaction between the following functions:

- The **Exponential potential** [8]: $\Gamma^t := \sum_{i=1}^{n} e^{\alpha(x_i^t - t/n)} + \sum_{i=1}^{n} e^{-\alpha(x_i^t - t/n)}$.
- The **Absolute potential**: $\Delta^t := \sum_{i=1}^{n} \left| x_i^t - t/n \right|$.
- The **Quadratic potential**: $\Upsilon^t := \sum_{i=1}^{n} \left(x_i^t - t/n \right)^2$.
- The **Quantile position**: $\delta^t = \left| \{ i \in [n] \colon y_i^t \geq 0 \} \right| / n$.

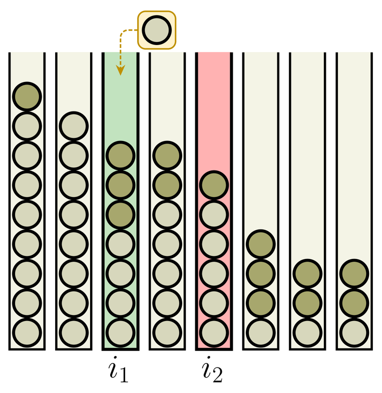
Rough idea (cf. [7]):

- Observe $e^{\text{Gap}(t)} \leq \Gamma^t$. *Want:* $\Gamma^t \in \text{Poly}(n)$ w.h.p., *Easy to get:* $\Gamma^t \leq e^{n \log n}$ w.h.p..
- As long as $\Delta^t = \Omega(n)$, then Υ^t drops in expectation.
- If $\Delta^t = \mathcal{O}(n)$, then $\delta^t \in (\varepsilon, 1 - \varepsilon)$ w.h.p., for a constant fraction of the next $\Theta(n)$ steps.
- If $\delta^t \in (\varepsilon, 1 - \varepsilon)$ then, in expectation, Γ^t decreases in the next step.
- Once $\Gamma^{t'} \leq cn$ then, w.h.p., for the next n^4 steps, it must be $\leq cn$ once every $n \log n$ steps.
- Between these events, w.h.p., the gap cannot rise by more than $\mathcal{O}(\log n)$.



Batching (3)

Two-Choice where balls are allocated in batches of size b.

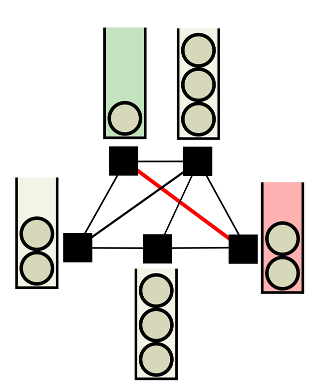


Recent balls (◯) are ignored; So, the ball is allocated to the heavier bin.

For $b \geq n \log n$, $\text{Gap}(m) = \Theta(b/n)$ [4].
For $b = n$, $\text{Gap}(m) = \Theta(\frac{\log n}{\log \log n})$ [6].

Graphical Allocations

Two-Choice on graphs: Sample an edge and allocate in the lesser loaded of the two bins.



On expanders, we show $\text{Gap}(m) = O(\log n)$ with weighted balls and batching [4].
On dense expanders, $\text{Gap}(m) = O(\log \log n)$ [5].

References

[1] Y. Azar, A. Z. Broder, A. R. Karlin, and E. Upfal. Balanced allocations. *SIAM J. Comput.*, 1999.

[2] P. Berenbrink, A. Czumaj, A. Steger, and B. Vöcking. Balanced allocations: the heavily loaded case. *SIAM J. Comput.*, 2006.

[3] R. M. Karp, M. Luby, and F. Meyer auf der Heide. Efficient PRAM simulation on a distributed memory machine. *Algorithmica*, 1996.

[4] D. Los and T. Sauerwald. Balanced Allocations in Batches: Simplified and Generalized. *SPAA 2022.*

[5] D. Los and T. Sauerwald. Balanced Allocations with Incomplete Information: The Power of Two Queries. *ITCS 2022.*

[6] D. Los and T. Sauerwald. Balanced Allocations with the Choice of Noise. *PODC 2022.*

[7] D. Los, T. Sauerwald, and J. Sylvester. Balanced Allocations: Caching and Packing, Twinning and Thinning. *SODA 2022.*

[8] Yuval Peres, Kunal Talwar, and Udi Wieder. Graphical balanced allocations and the $(1 + \beta)$-choice process. *Random Structures Algorithms*, 2015.

[9] M. Raab and A. Steger. "Balls into bins" - a simple and tight analysis. *RANDOM 1998.*