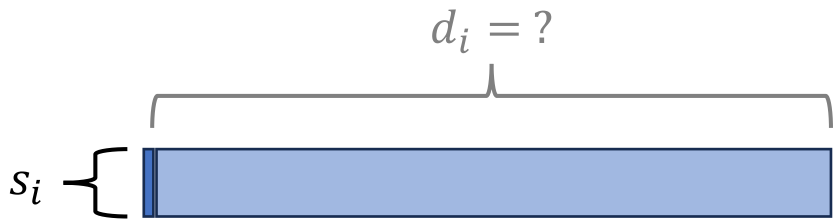
The Power of Migrations in Dynamic Bin Packing

Konstantina Mellou Marco Molinaro **Rudy Zhou**

Microsoft Research Carnegie Mellon ⇒ Microsoft

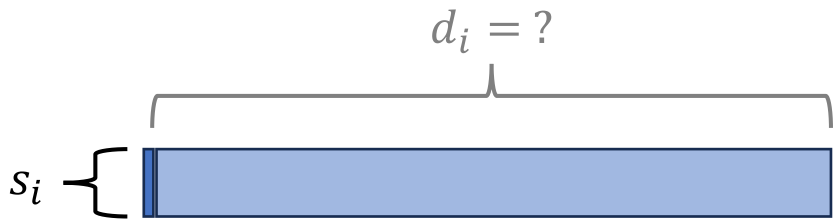
To appear in Sigmetrics 2025

Dynamic Bin Packing



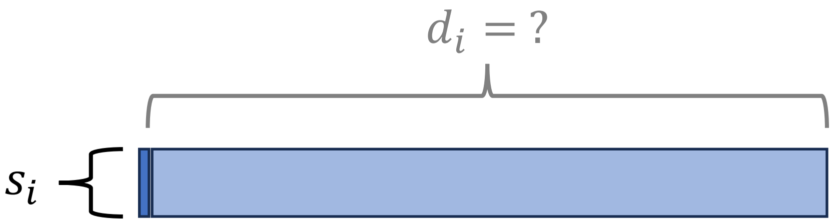
- Items arrive online at their arrival time with sizes
- Items depart after their (unknown) duration
- Must pack into unit-size bins

Dynamic Bin Packing

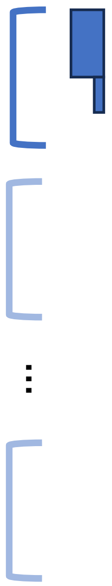


- Items arrive online at their arrival time with sizes
- Items depart after their (unknown) duration
- Must pack into unit-size bins

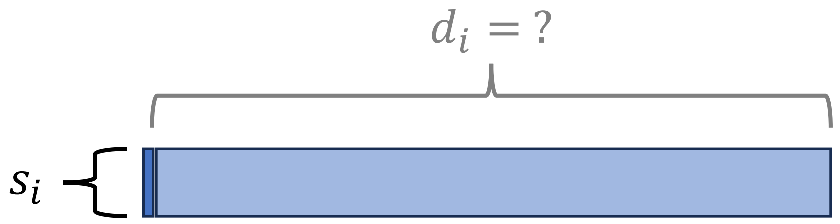
Dynamic Bin Packing



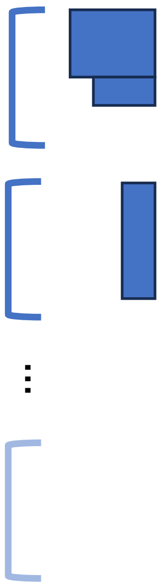
- Items arrive online at their arrival time with sizes
- Items depart after their (unknown) duration
- Must pack into unit-size bins



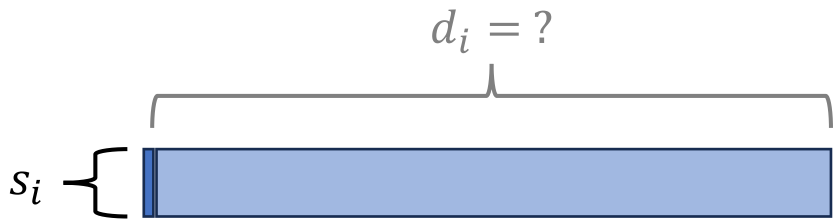
Dynamic Bin Packing



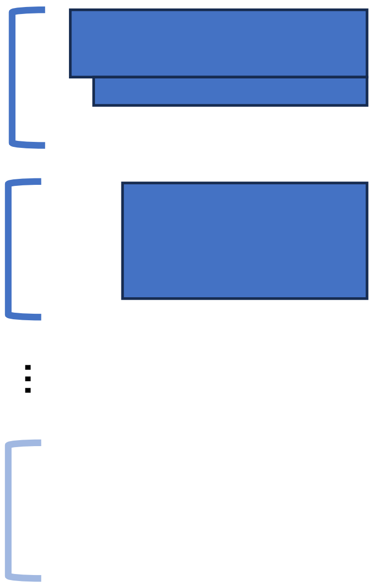
- Items arrive online at their arrival time with sizes
- Items depart after their (unknown) duration
- Must pack into unit-size bins



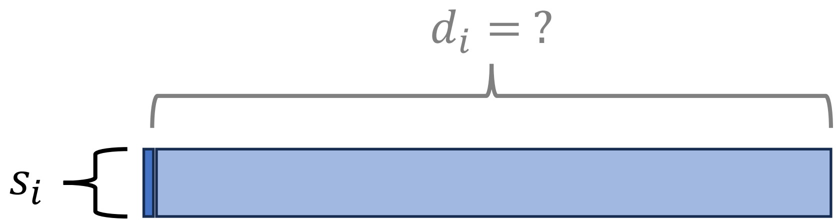
Dynamic Bin Packing



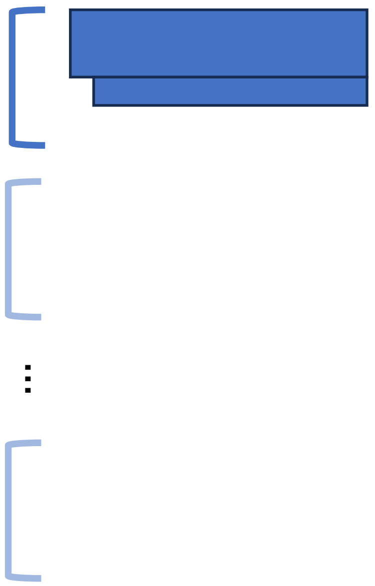
- Items arrive online at their arrival time with sizes
- Items depart after their (unknown) duration
- Must pack into unit-size bins



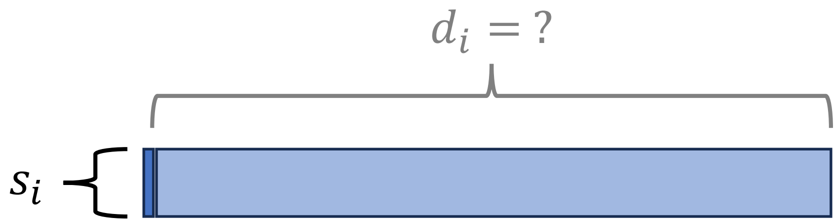
Dynamic Bin Packing



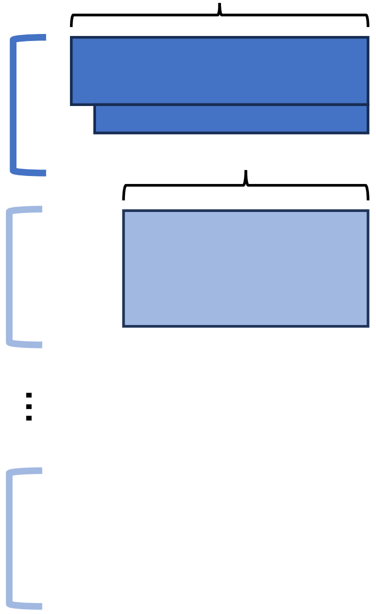
- Items arrive online at their arrival time with sizes
- Items depart after their (unknown) duration
- Must pack into unit-size bins



Dynamic Bin Packing

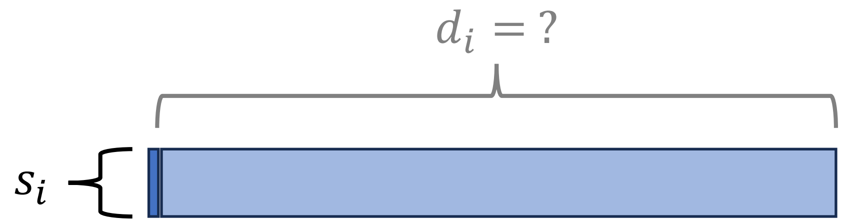


- Items arrive online at their arrival time with sizes
- Items depart after their (unknown) duration
- Must pack into unit-size bins

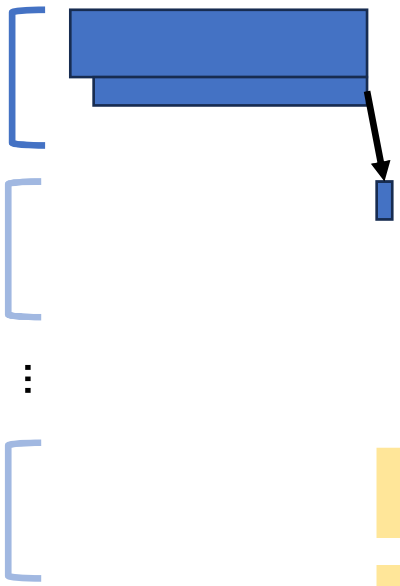


Minimize total active time over all bins

Dynamic Bin Packing



- Items arrive online at their arrival time with sizes
- Items depart after their (unknown) duration
- Must pack into unit-size bins



Minimize total active time over all bins

Can also migrate items

Related Work

$$\mu = \frac{\max duration}{\min duration}$$

$$n = \#\, items$$

- **No migrations:** $\Theta(\mu)$-approximation (first fit)

  Yusen Li, Xueyan Tang, Wentong Cai:
  On dynamic bin packing for resource allocation in the cloud. SPAA 2014

- $> \boldsymbol{n}$ **migrations:** $\approx 1.387 + \epsilon$-approximation with $O(\frac{n}{\epsilon^2})$ migrations

  Björn Feldkord, Matthias Feldotto, Anupam Gupta, Guru Guruganesh, Amit Kumar, Sören Riechers, David Wajc:
  Fully-Dynamic Bin Packing with Little Repacking. ICALP 2018

- Can get better guarantees with no migrations if know item durations exactly or approximately (predictions)

  Yossi Azar, Danny Vainstein:
  Tight Bounds for Clairvoyant Dynamic Bin Packing. SPAA 2017

  Mozhengfu Liu, Xueyan Tang:
  Dynamic Bin Packing with Predictions. SIGMETRICS 2023

  Niv Buchbinder, Yaron Fairstein, Konstantina Mellou, Ishai Menache, Joseph (Seffi) Naor:
  Online Virtual Machine Allocation with Lifetime and Load Predictions. SIGMETRICS 2021

Related Work

$$\mu = \frac{\max duration}{\min duration}$$

$$n = \#\ items$$

- **No migrations:** $\Theta(\mu)$-approximation (first fit)

  Yusen Li, Xueyan Tang, Wentong Cai:
  On dynamic bin packing for resource allocation in the cloud. SPAA 2014

- $> \boldsymbol{n}$ **migrations:** $\approx 1.387 + \epsilon$-approximation with $O(\frac{n}{\epsilon^2})$ migrations

  Björn Feldkord, Matthias Feldotto, Anupam Gupta, Guru Guruganesh, Amit Kumar, Sören Riechers, David Wajc:
  Fully-Dynamic Bin Packing with Little Repacking. ICALP 2018

- Can get better guarantees with no migrations if know item durations exactly or approximately (predictions)

  Yossi Azar, Danny Vainstein:
  Tight Bounds for Clairvoyant Dynamic Bin Packing. SPAA 2017

  Mozhengfu Liu, Xueyan Tang:
  Dynamic Bin Packing with Predictions. SIGMETRICS 2023

  Niv Buchbinder, Ya
  Online Virtual Mac

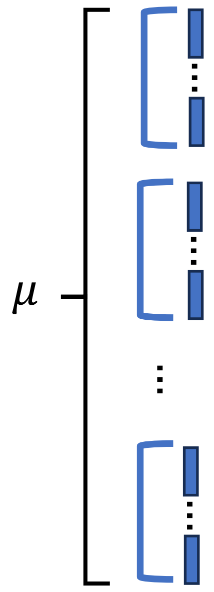
What can we do with $< n$ migrations? ϵn? $\sqrt{n}$?

Bad Example

- μ^2 items; all arriving at time 0 with size $\frac{1}{\mu}$
  - μ of them are long with duration μ
  - Rest are short with duration 1

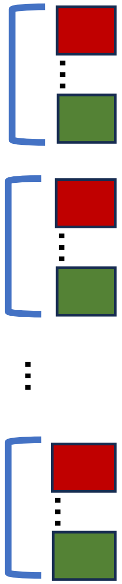
Bad Example

- μ^2 items; all arriving at time 0 with size $\frac{1}{\mu}$
  - μ of them are long with duration μ
  - Rest are short with duration 1



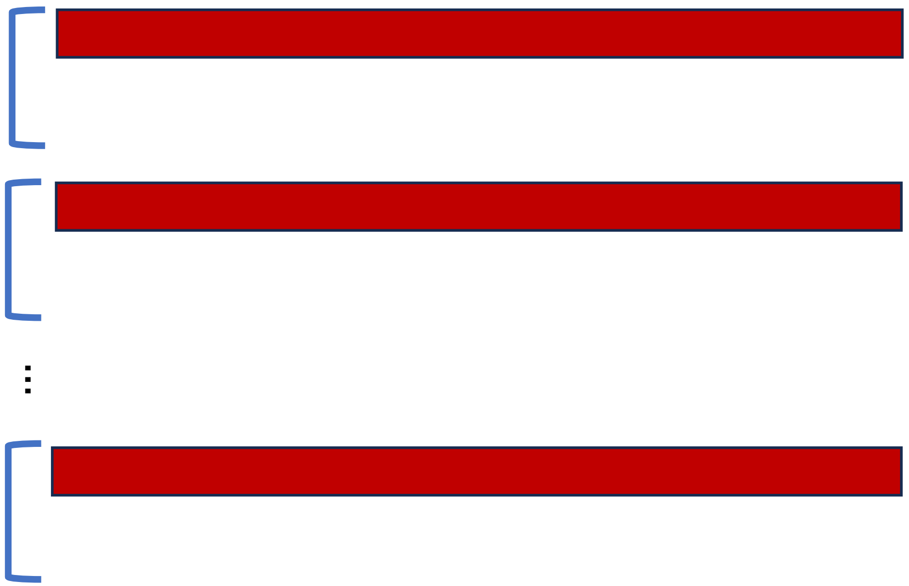
Bad Example

- μ^2 items; all arriving at time 0 with size $\frac{1}{\mu}$
  - μ of them are long with duration μ
  - Rest are short with duration 1



Bad Example

- μ^2 items; all arriving at time 0 with size $\frac{1}{\mu}$
  - μ of them are long with duration μ
  - Rest are short with duration 1

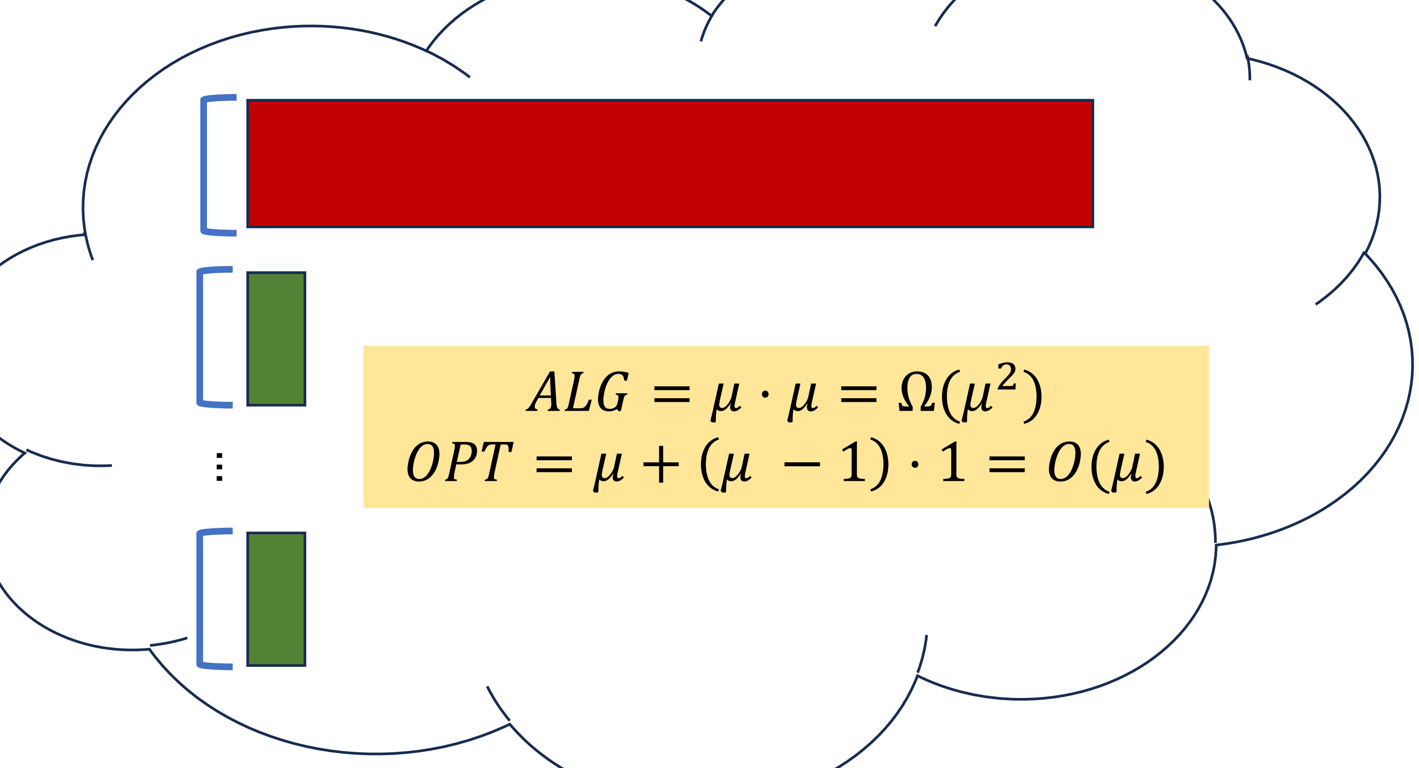


Bad Example

- μ^2 items; all arriving at time 0 with size $\frac{1}{\mu}$
  - μ of them are long with duration μ
  - Rest are short with duration 1

Bad Example

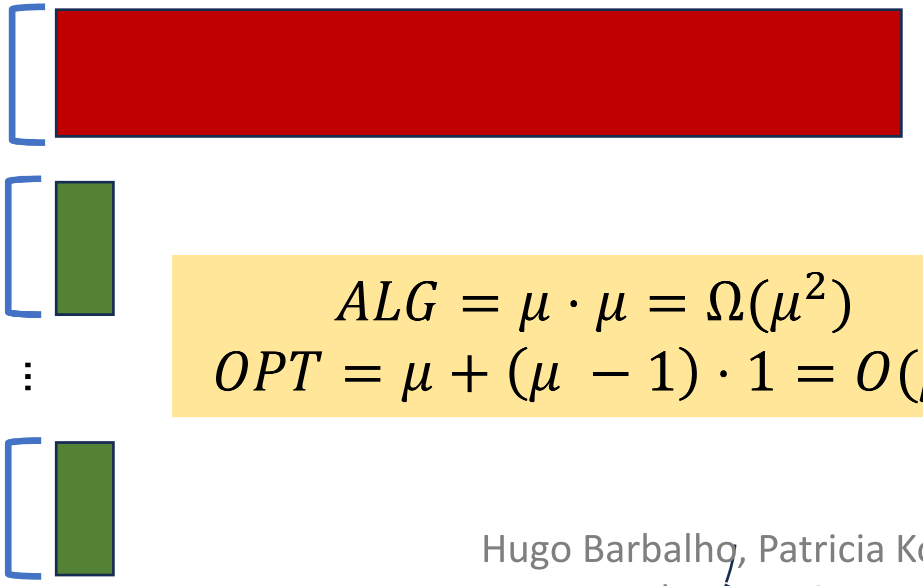
- μ^2 items; all arriving at time 0 with size $\frac{1}{\mu}$
  - μ of them are long with duration μ
  - Rest are short with duration 1



$$ALG = \mu \cdot \mu = \Omega(\mu^2)$$
$$OPT = \mu + (\mu - 1) \cdot 1 = O(\mu)$$

Bad Example

- μ^2 items; all arriving at time 0 with size $\frac{1}{\mu}$
  - μ of them are long with duration μ
  - Rest are short with duration 1



$$ALG = \mu \cdot \mu = \Omega(\mu^2)$$
$$OPT = \mu + (\mu - 1) \cdot 1 = O(\mu)$$

This actually happens in practice

Hugo Barbalho, Patricia Kovaleski, Beibin Li, Luke Marshall, Marco Molinaro, Abhisek Pan, Eli Cortez, Matheus Leao, Harsh Patwari, Zuzu Tang, Larissa Rozales Gonçalves, David Dion, Thomas Moscibroda, Ishai Menache: Virtual Machine Allocation with Lifetime Predictions. MLSys 2023

Our Results (Part 1)

- **Sublinear migrations:** Any algorithm that does $o(n)$ migrations must be $\Omega(\mu)$-approximate
- $< \boldsymbol{n}$ **migrations:** For any $\epsilon \in (0,1)$, can get $\approx \frac{1}{\epsilon}$-approximation using ϵn migrations, and this is best possible*

* up to a $\log n$ additive term in approximation

Our Results (Part 1)

- **Sublinear migrations:** Any algorithm that does $o(n)$ migrations must be $\Omega(\mu)$-approximate
- **$< \boldsymbol{n}$ migrations:** For any $\epsilon \in (0,1)$, can get $\approx \frac{1}{\epsilon}$-approximation using ϵn migrations, and this is best possible*
- **$> \boldsymbol{n}$ migrations:** $\approx 1.387 + \epsilon$-approximation with $O(\frac{n}{\epsilon^2})$ migrations

* up to a $\log n$ additive term in approximation

$< n$ migrations

- Want $\approx \frac{1}{\epsilon}$-approximation $\Rightarrow$ suffices to ensure most bins are $\geq \epsilon$-full
- Assume all items have same size*

* standard discretization argument; lose additive $\log n$ here

$< n$ migrations

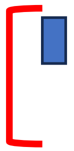
- Want $\approx \frac{1}{\epsilon}$-approximation $\Rightarrow$ suffices to ensure most bins are $\geq \epsilon$-full
- Assume all items have same size*
- Classify bins as bad and good

* standard discretization argument; lose additive $\log n$ here

$< n$ migrations

- Want $\approx \frac{1}{\epsilon}$-approximation $\Rightarrow$ suffices to ensure most bins are $\geq \epsilon$-full
- Assume all items have same size*
- Classify bins as bad and good

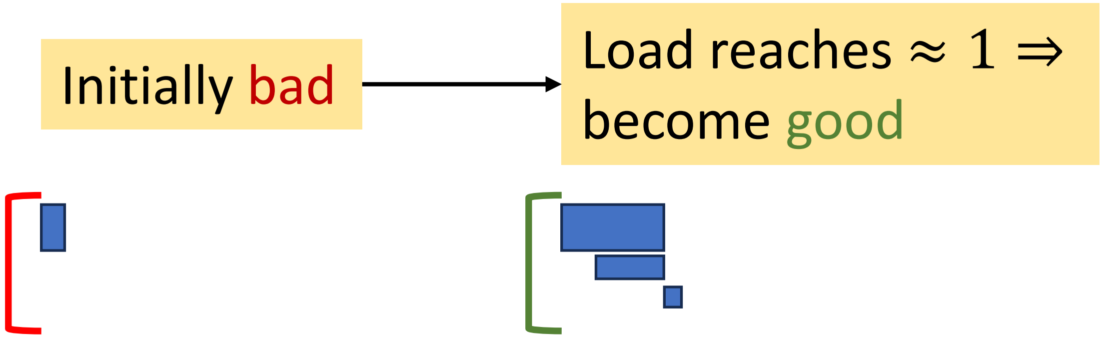
Initially bad



* standard discretization argument; lose additive $\log n$ here

$< n$ migrations

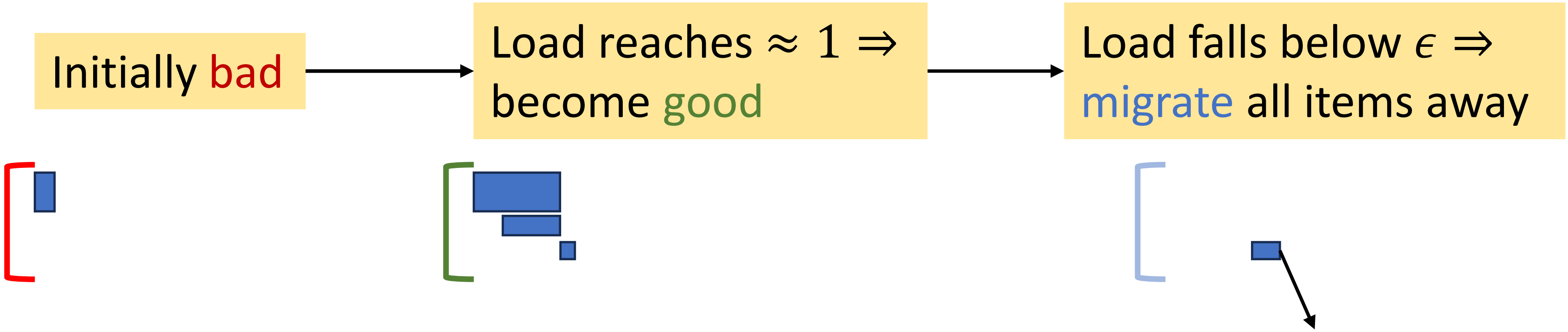
- Want $\approx \frac{1}{\epsilon}$-approximation $\Rightarrow$ suffices to ensure most bins are $\geq \epsilon$-full
- Assume all items have same size*
- Classify bins as bad and good



* standard discretization argument; lose additive $\log n$ here

$< n$ migrations

- Want $\approx \frac{1}{\epsilon}$-approximation $\Rightarrow$ suffices to ensure most bins are $\geq \epsilon$-full
- Assume all items have same size*
- Classify bins as bad and good



* standard discretization argument; lose additive $\log n$ here

$< n$ migrations

- Want $\approx \frac{1}{\epsilon}$-approximation $\Rightarrow$ suffices to ensure most bins are $\geq \epsilon$-full
- Assume all items have same size*
- Classify bins as bad and good

Initially bad → Load reaches $\approx 1 \Rightarrow$ become good → Load falls below $\epsilon \Rightarrow$ migrate all items away

- Can ensure ≤ 1 bad bins at any time
- Migrate ϵ-fraction of bin load when $1 - \epsilon$-fraction departs

* standard discretization argument; lose additive $\log n$ here

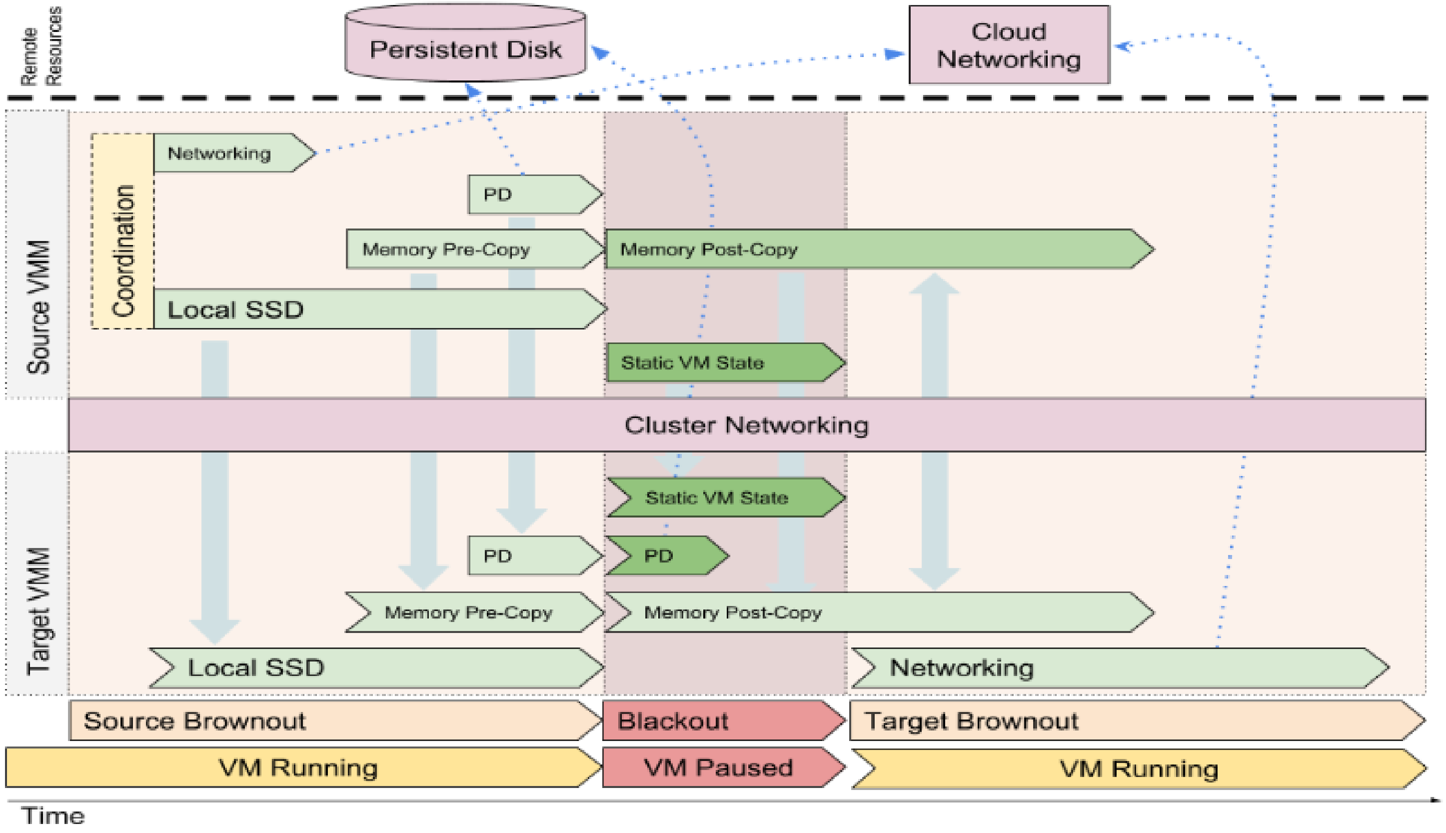
Our Results (Part 1)

- **Sublinear migrations:** Any algorithm that does $o(n)$ migrations must be $\Omega(\mu)$-approximate
- **$< n$ migrations:** For any $\epsilon \in (0, 1)$, can get $\approx \frac{1}{\epsilon}$-approximation using ϵn migrations, and this is best possible*
- **$> n$ migrations:** $\approx 1.387 + \epsilon$-approximation with $O(\frac{n}{\epsilon^2})$ migrations

Can we do better?

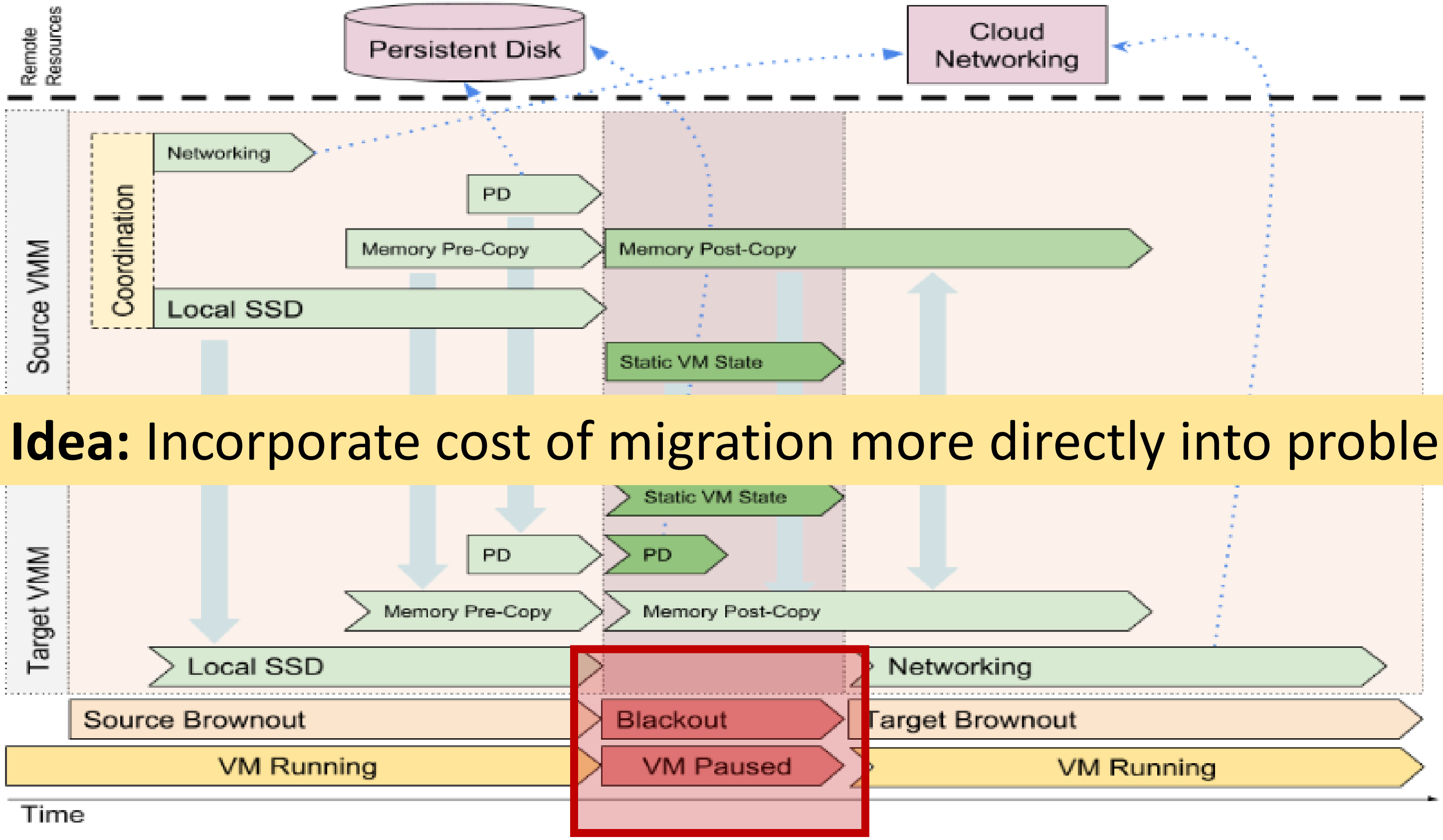
* up to a $\log n$ additive term in approximation

Real cost of migration



Adam Ruprecht, Danny Jones, Dmitry Shiraev, Greg Harmon, Maya Spivak, Michael Krebs, Miche Baker-Harvey, Tyler Sanderson: VM Live Migration At Scale. VEE 2018

Real cost of migration

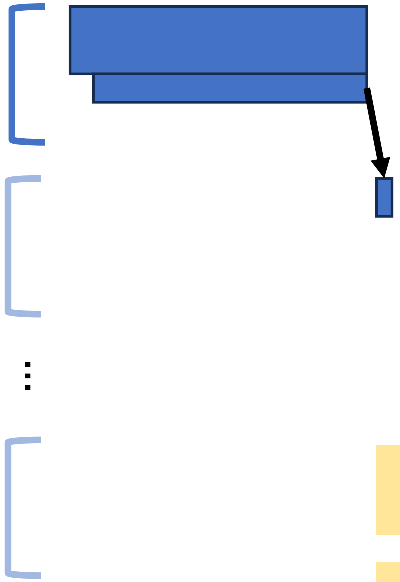


Idea: Incorporate cost of migration more directly into problem

Adam Ruprecht, Danny Jones, Dmitry Shiraev, Greg Harmon, Maya Spivak, Michael Krebs, Miche Baker-Harvey, Tyler Sanderson: VM Live Migration At Scale. VEE 2018

Dynamic Bin Packing with Delays

- Items arrive online at their arrival time with sizes
- Items depart after their (unknown) duration
- Must pack into unit-size bins

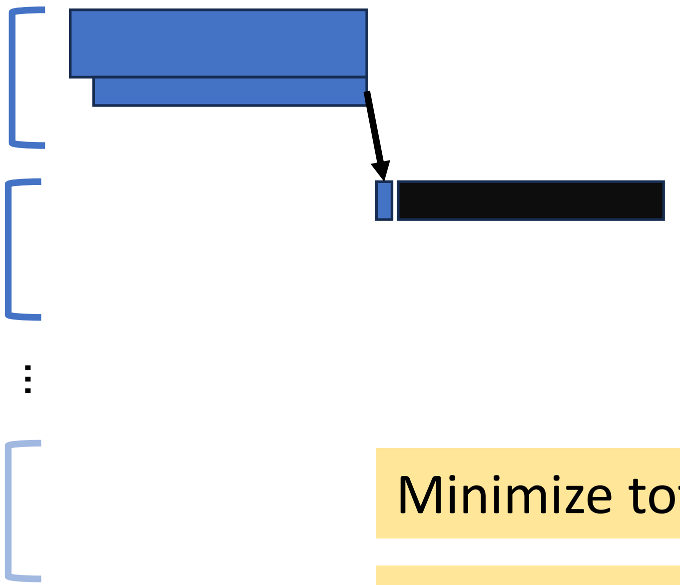


Minimize total active time over all bins

Can also migrate items

Dynamic Bin Packing with Delays

- Items arrive online at their arrival time with sizes
- Items depart after their (unknown) duration
- Must pack into unit-size bins

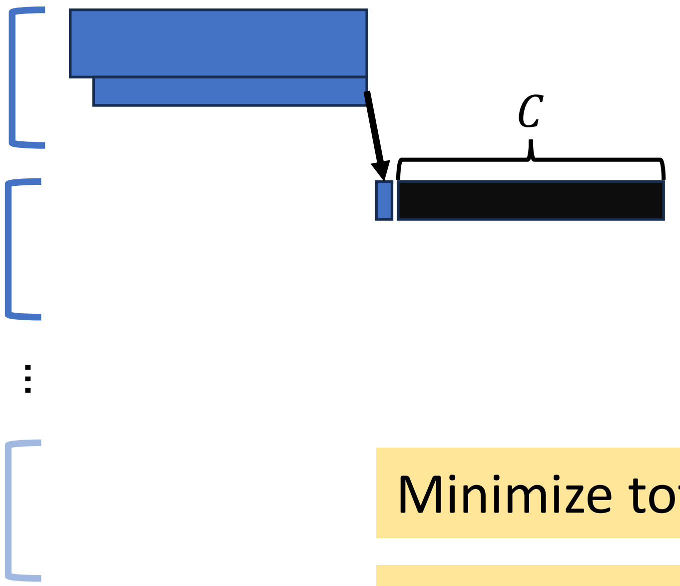


Minimize total active time over all bins

Can also migrate items, but migrating increases item duration by $+C$

Dynamic Bin Packing with Delays

- Items arrive online at their arrival time with sizes
- Items depart after their (unknown) duration
- Must pack into unit-size bins

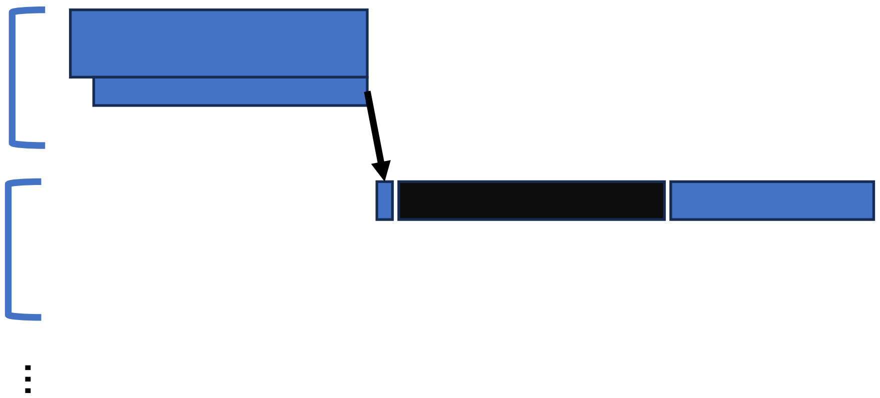


Minimize total active time over all bins

Can also migrate items, but migrating increases item duration by $+C$

Dynamic Bin Packing with Delays

- Items arrive online at their arrival time with sizes
- Items depart after their (unknown) duration
- Must pack into unit-size bins



Minimize total active time over all bins

Can also migrate items, but migrating increases item duration by $+C$

Our Results (Part 2)

- $O(\min(\sqrt{C}, \mu))$-approximation for Dynamic bin packing with delays, and this is best possible*

* some normalization assumptions apply

Our Results (Part 2)

- $O(\min(\sqrt{C}, \mu))$-approximation for Dynamic bin packing with delays, and this is best possible*
- Best of both worlds: never worse than doing no migrations, but can be much better

* some normalization assumptions apply

Our Results (Part 2)

- $O(\min(\sqrt{C}, \mu))$-approximation for Dynamic bin packing with delays, and this is best possible*
- Best of both worlds: never worse than doing no migrations, but can be much better
- In practice, the minimum duration can be ≈ 1 millisecond, and the maximum ≈ 1 year ⇒ $\mu \approx 10^{10}$
- However, migrating an item incurs a delay of ≈ 1 second ⇒ $C \approx 10^3$

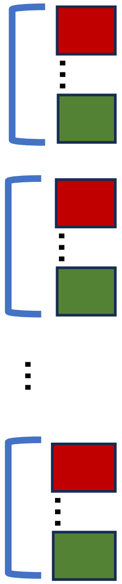
* some normalization assumptions apply

Dynamic Bin Packing with Delays

- When to migrate?
- Recall bad example: all items have same size, some short, some long

Dynamic Bin Packing with Delays

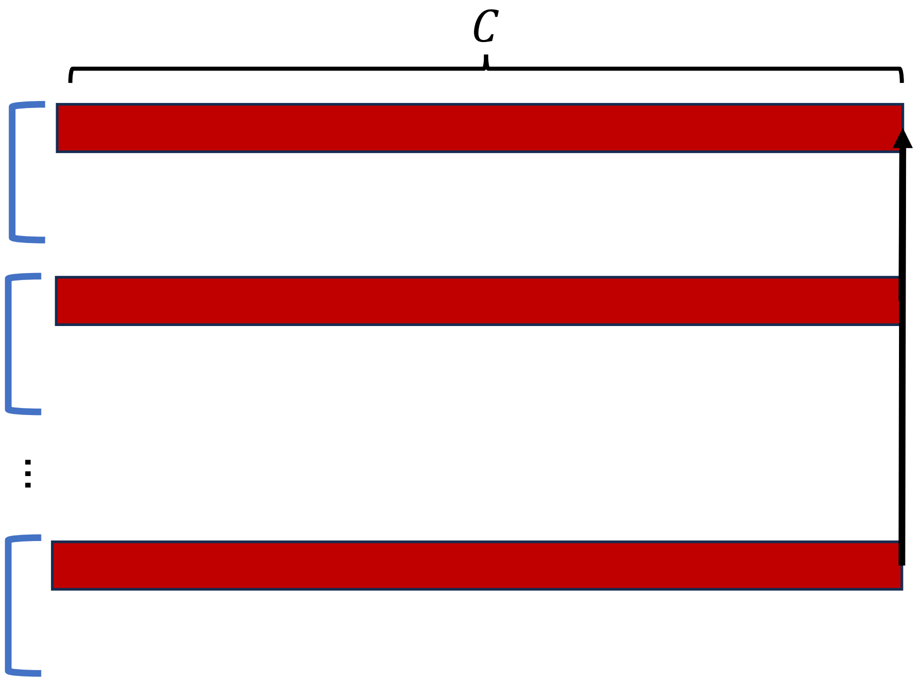
- When to migrate?
- Recall bad example: all items have same size, some short, some long



Dynamic Bin Packing with Delays

First Try: Migrate at time C

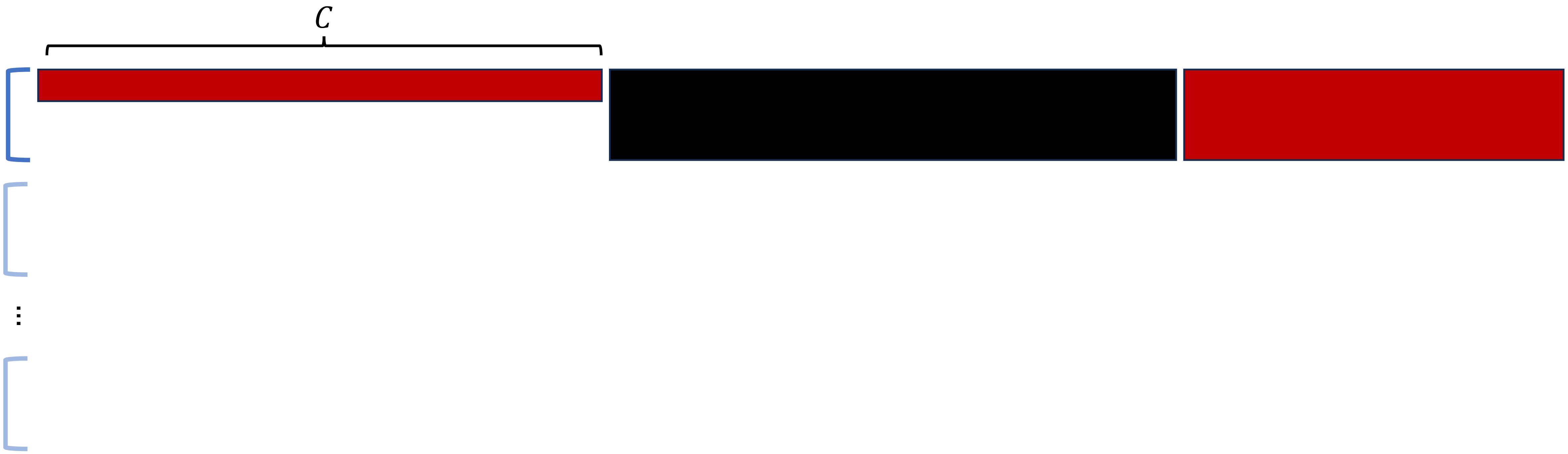
- When to migrate?
- Recall bad example: all items have same size, some short, some long



Dynamic Bin Packing with Delays

First Try: Migrate at time C

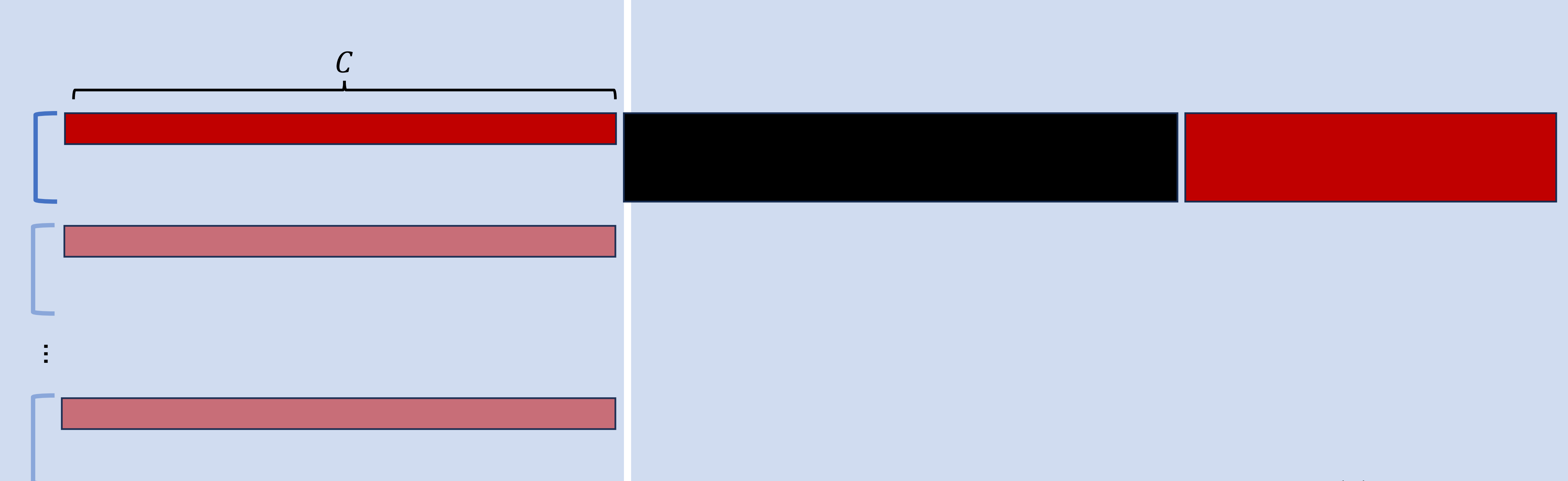
- When to migrate?
- Recall bad example: all items have same size, some short, some long



Dynamic Bin Packing with Delays

First Try: Migrate at time C

- When to migrate?
- Recall bad example: all items have same size, some short, some long



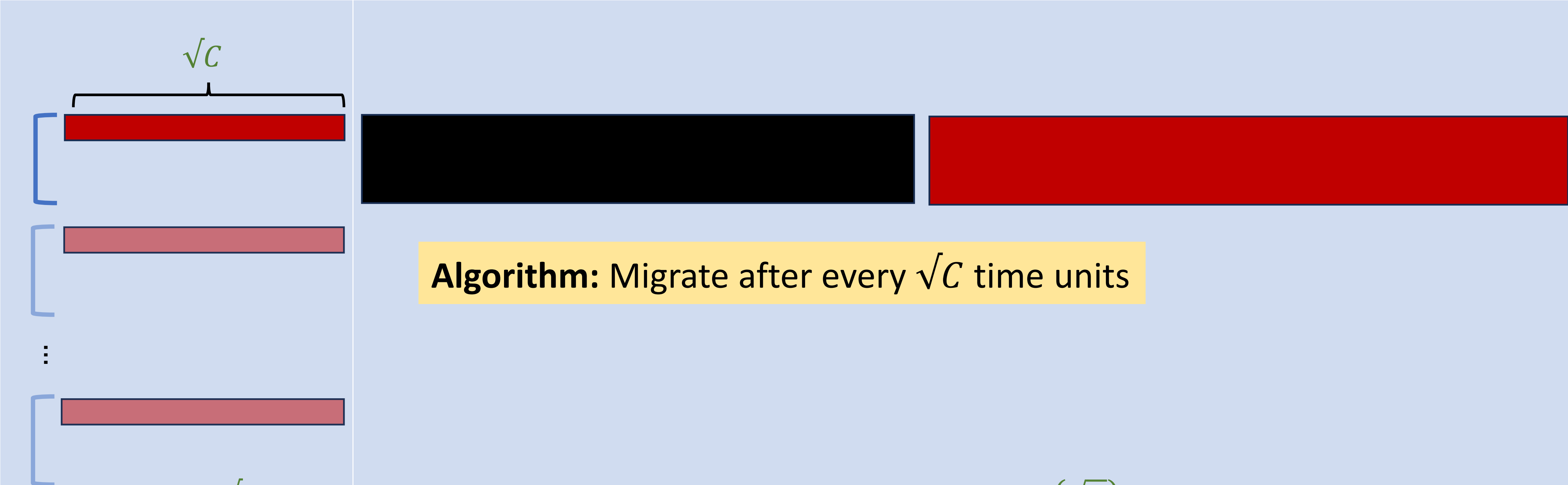
Bad assignment for C time units

Good assignment, but increased durations by $O(1)$-factor

Dynamic Bin Packing with Delays

First Try: Migrate at time C

- When to migrate?
- Recall bad example: all items have same size, some short, some long



Bad assignment for $\sqrt{C}$ time units

Good assignment, but increased durations by $O(\sqrt{C})$-factor

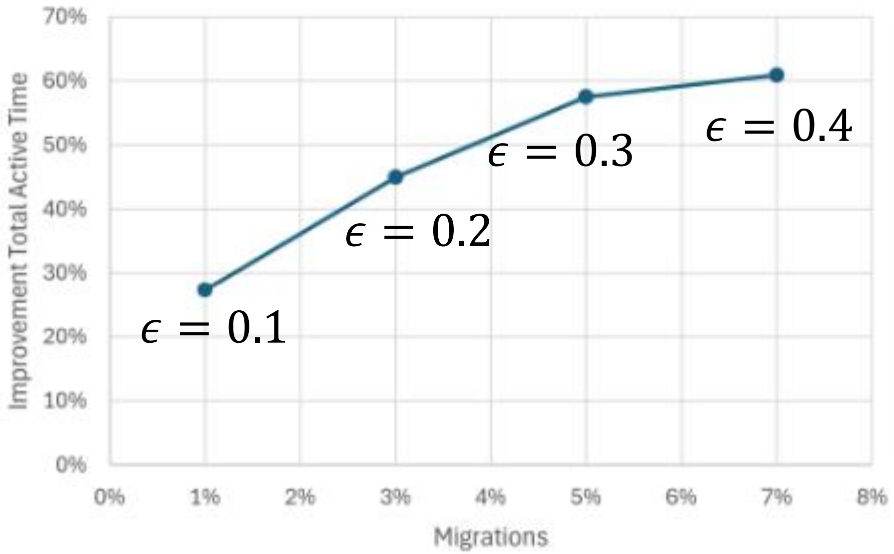
Our Results (Part 2)

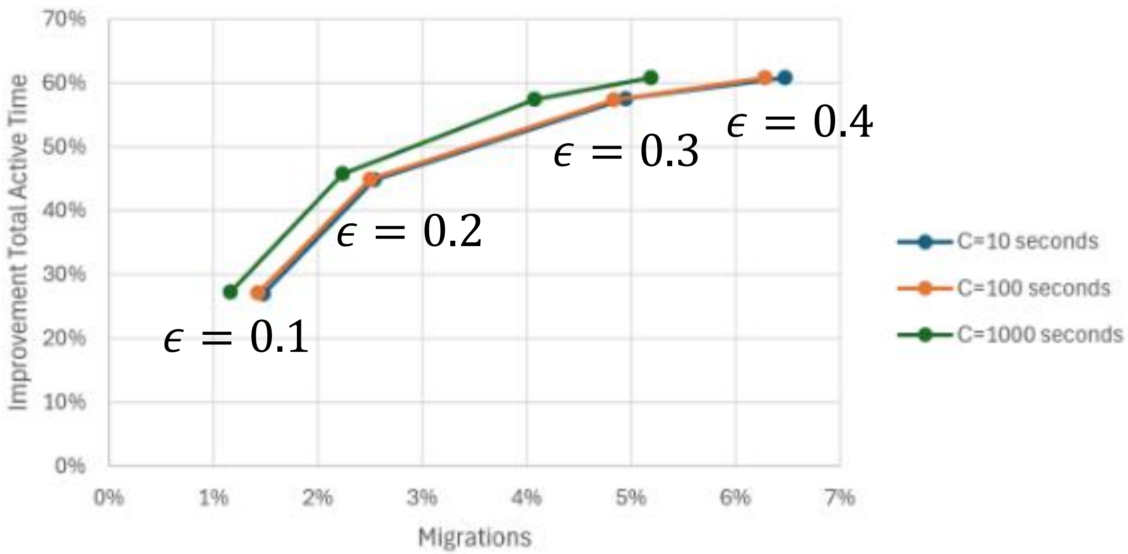
- $O(\min(\sqrt{C}, \mu))$-approximation for Dynamic bin packing with delays, and this is best possible*
- Best of both worlds: never worse than doing no migrations, but can be much better
- In practice, the minimum duration can be ≈ 1 millisecond, and the maximum ≈ 1 year ⇒ $\mu \approx 10^{10}$
- However, migrating an item incurs a delay of ≈ 1 second ⇒ $C \approx 10^3$

* some normalization assumptions apply

Experiments

- Data from Microsoft Azure (VM requests)
- Combine both $< n$ migration and delay algorithms (classify bins as bad and good; only migrate items after every C time units)

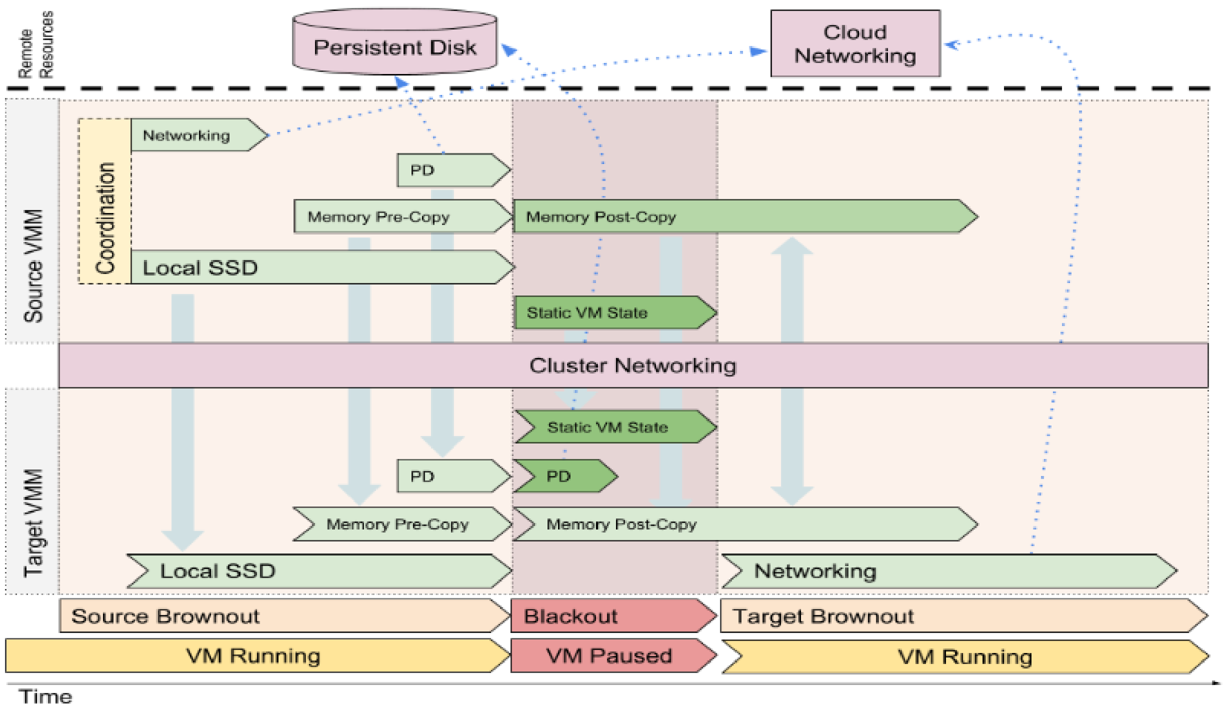




Ori Hadary, Luke Marshall, Ishai Menache, Abhisek Pan, Esaias E. Greeff, David Dion, Star Dorminey, Shailesh Joshi, Yang Chen, Mark Russinovich, Thomas Moscibroda: Protean: VM Allocation Service at Scale. OSDI 2020

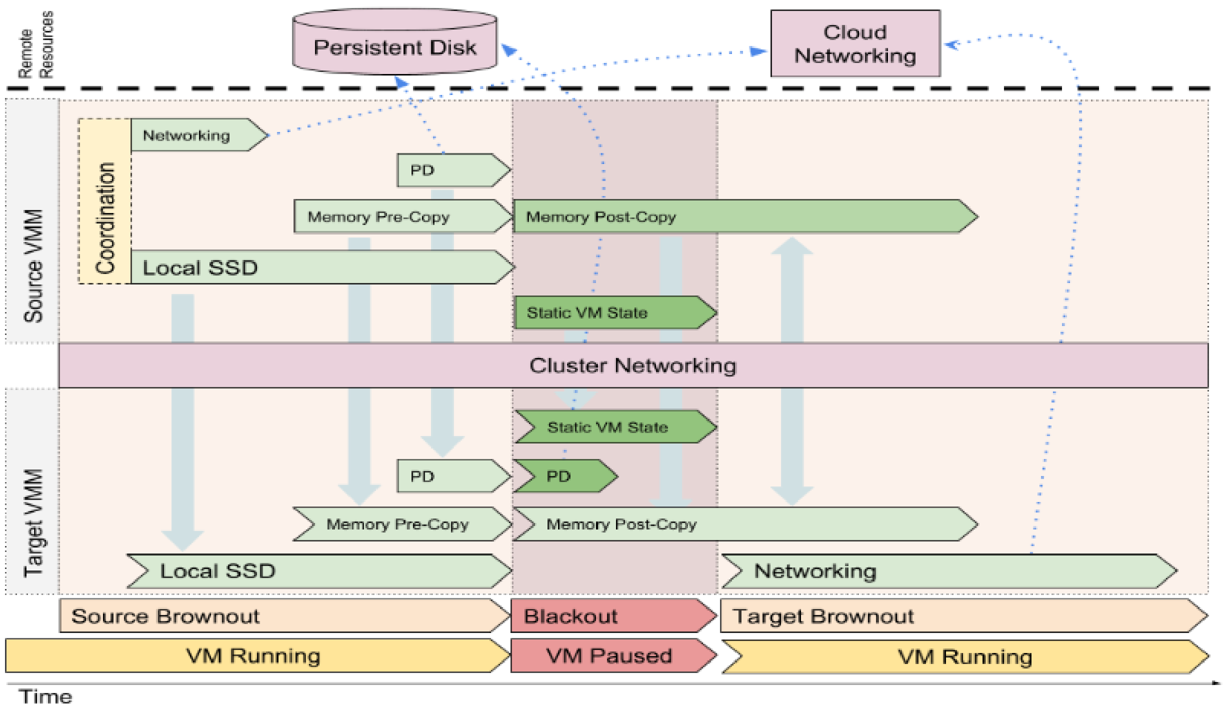
Conclusion

- Fill gaps in our understanding for $o(n)$ and ϵn migrations
- Introduce delays to dynamic bin packing



Conclusion

- Fill gaps in our understanding for $o(n)$ and ϵn migrations
- Introduce delays to dynamic bin packing



Open Questions:

- Remove additive $\log n$ in ϵn migration case
- Beyond worst case model
- Consider networking limits