Algorithms for Dynamic Right-Sizing in Data Centers

Kevin Kappelmann

July 24, 2017

Technical University of Munich

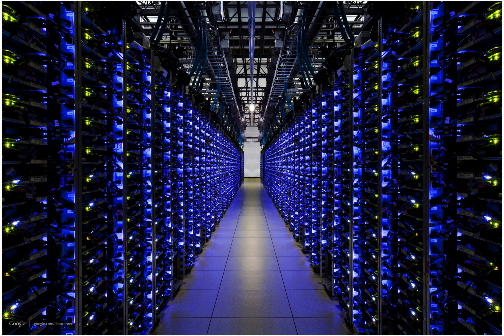
Motivation and Problem Statement

Data Centers' Costs

Fast-growing demand for data collection, processing, and storage

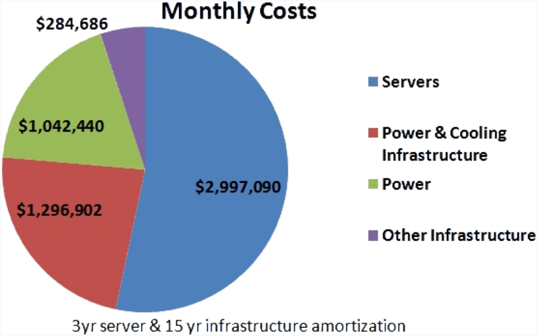
Data Centers' Costs

Fast-growing demand for data collection, processing, and storage



Data Centers' Costs

Fast-growing demand for data collection, processing, and storage



Our Model

- $m \in \mathbb{N}$ ... Number of homogeneous servers

Our Model

- $m \in \mathbb{N} \ldots$ Number of homogeneous servers
- $T \in \mathbb{N} \ldots$ Number of time slots

Our Model

- $m \in \mathbb{N}$ ... Number of homogeneous servers
- $T \in \mathbb{N}$ ... Number of time slots
- $\beta \in \mathbb{R}_{\geq 0}$ ... Switching costs of a server

Our Model

- $m \in \mathbb{N}$ ... Number of homogeneous servers
- $T \in \mathbb{N}$ ... Number of time slots
- $\beta \in \mathbb{R}_{\geq 0}$ ... Switching costs of a server
- $f : [0,1] \to \mathbb{R}$ ... Convex operating cost function of a server

Our Model

- $m \in \mathbb{N}$ ... Number of homogeneous servers
- $T \in \mathbb{N}$ ... Number of time slots
- $\beta \in \mathbb{R}_{\geq 0}$ ... Switching costs of a server
- $f : [0, 1] \to \mathbb{R}$ ... Convex operating cost function of a server
- $\lambda_1, \dots, \lambda_T \in [0, m]$ ... Arrival rates

Our Model

- $m \in \mathbb{N}$ ... Number of homogeneous servers
- $T \in \mathbb{N}$ ... Number of time slots
- $\beta \in \mathbb{R}_{\geq 0}$ ... Switching costs of a server
- $f : [0, 1] \to \mathbb{R}$ ... Convex operating cost function of a server
- $\lambda_1, \dots, \lambda_T \in [0, m]$ ... Arrival rates
- $x_1, \dots, x_T \in \{0, \dots, m\}$ ... Numbers of active servers

Problem Statement

Operating costs for one time step

$$c_{op}(x_t, \lambda_t) := \begin{cases} \infty, & \text{if } \lambda_t > x_t \quad \text{//too few servers} \\ x_t f(\lambda_t / x_t), & \text{if } x_t \neq 0 \land \lambda_t \leq x_t \quad \text{//even distribution} \\ 0, & \text{if } x_t = \lambda_t = 0 \quad \text{//no active servers} \end{cases}$$

Problem Statement

Operating costs for one time step

$$c_{op}(x_t, \lambda_t) := \begin{cases} \infty, & \text{if } \lambda_t > x_t & \text{//too few servers} \\ x_t f(\lambda_t / x_t), & \text{if } x_t \neq 0 \wedge \lambda_t \leq x_t & \text{//even distribution} \\ 0, & \text{if } x_t = \lambda_t = 0 & \text{//no active servers} \end{cases}$$

Goal: Minimize total costs

$$\text{minimize} \quad \sum_{t=1}^{T} \Big(\underbrace{c_{op}(x_t, \lambda_t) + \beta \max\{0, x_t - x_{t-1}\}}_{c(x_{t-1}, x_t, \lambda_t)} \Big)$$

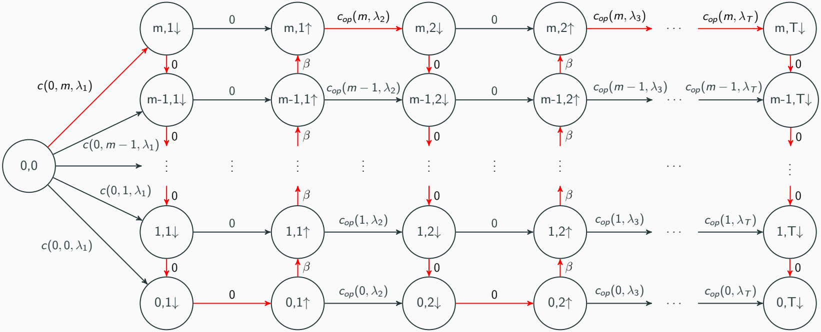
Optimal Offline Algorithm

Pseudo-Linear-Time Algorithm

Fundamental idea: reduce problem to shortest path problem

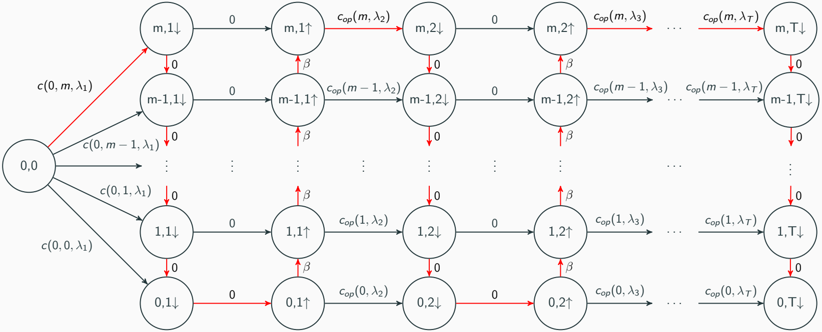
Pseudo-Linear-Time Algorithm

Fundamental idea: reduce problem to shortest path problem



Pseudo-Linear-Time Algorithm

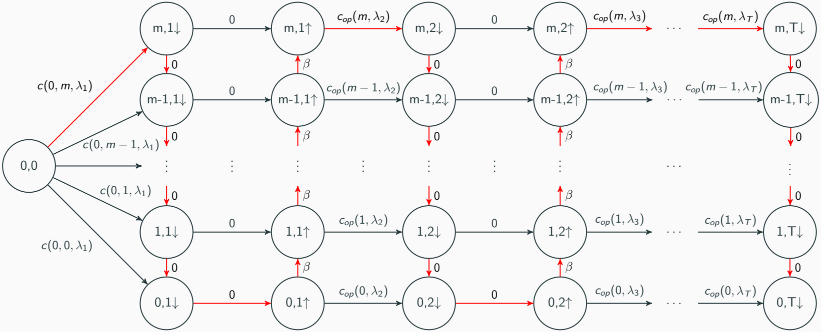
Fundamental idea: reduce problem to shortest path problem



Time complexity: $\Theta(Tm)$

Pseudo-Linear-Time Algorithm

Fundamental idea: reduce problem to shortest path problem



Time complexity: $\Theta(Tm)$

. . . only $\log_2(m)$ bits required to encode m.

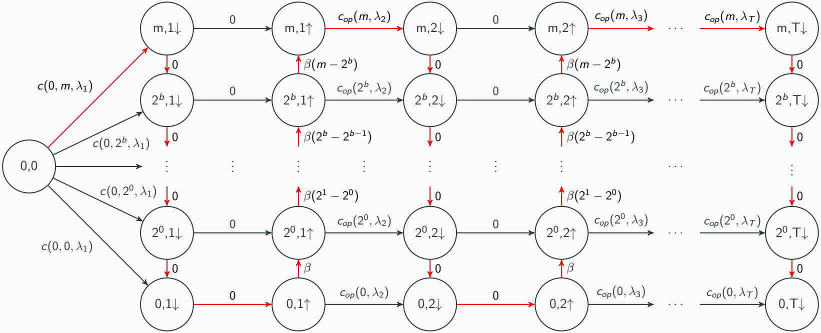
Offline Approximation Algorithm

2-Optimal Linear-Time Algorithm

Use logarithmic steps to reduce number of nodes

2-Optimal Linear-Time Algorithm

Use logarithmic steps to reduce number of nodes



Proof Sketch

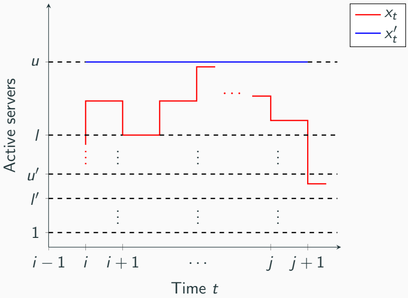
Transform periods between two powers of 2

Original schedule between $[l, u)$ where $l = 2^k, u = 2^{k+1}$

Proof Sketch

Transform periods between two powers of 2

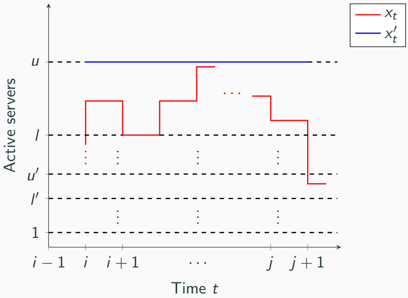
Original schedule between $[l, u)$ where $l = 2^k, u = 2^{k+1}$

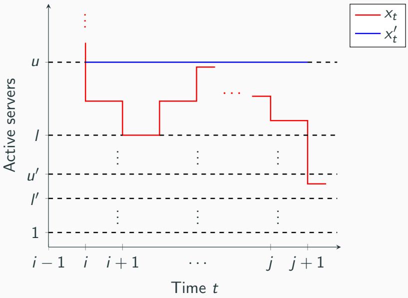


Proof Sketch

Transform periods between two powers of 2

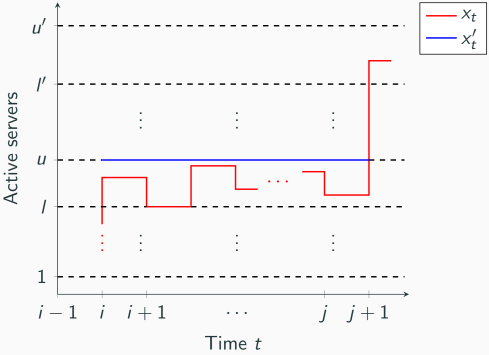
Original schedule between $[l, u)$ where $l = 2^k, u = 2^{k+1}$





Proof Sketch

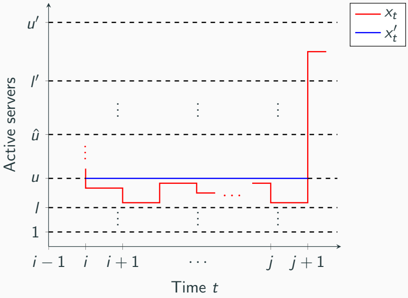
Original schedule between $[l, u)$ where $l = 2^k, u = 2^{k+1}$



Proof Sketch

Original schedule between $[l, u)$ where $l = 2^k, u = 2^{k+1}$

Decide between u and $\hat{u} = 2^{k+2}$



Approximative Scheduling

Shortest Path in graph corresponds to 2-optimal schedule.

Approximative Scheduling

Shortest Path in graph corresponds to 2-optimal schedule.

Time complexity: $\Theta\big(T \log_2(m)\big)$

Approximative Scheduling

Shortest Path in graph corresponds to 2-optimal schedule.

Time complexity: $\Theta\big(T \log_2(m)\big)$

Approach can be generalized to allow for arbitrary precisions with time complexity $\Theta\left(T\,\frac{\log(m)}{\log(1+\varepsilon)}\right)$

Summary and Prospects

Summary

We reduced the scheduling problem to the shortest path problem of acyclic graphs.

Summary

We reduced the scheduling problem to the shortest path problem of acyclic graphs.

$$\boxed{\text{Optimal Offline Algorithm with runtime } \Theta(Tm)}$$

Summary

We reduced the scheduling problem to the shortest path problem of acyclic graphs.

Optimal Offline Algorithm with runtime $\Theta(Tm)$

$(1+\varepsilon)$-Optimal Offline Algorithm with runtime
$\Theta\left(T\,\frac{\log(m)}{\log(1+\varepsilon)}\right)$

Prospects

Can approach be modified to . . .

Prospects

Can approach be modified to . . .

- deal with more than one homogeneous server collection?

Prospects

Can approach be modified to . . .

- deal with more than one homogeneous server collection?
- deal with multiple sleep states?

Prospects

Can approach be modified to . . .

- deal with more than one homogeneous server collection?
- deal with multiple sleep states?
- work as an online algorithm?

Prospects

Can approach be modified to . . .

- deal with more than one homogeneous server collection?
- deal with multiple sleep states?
- work as an online algorithm?

Open question: Is there a polynomial optimal algorithm or is it an **NP** problem?

Thanks for your attention!

Any questions?

Image Sources I

- Data center: datacentervoice.com/wp-content/uploads/2015/12/data-center.jpg
- Data center costs: perspectives.mvdirona.com/2008/11/cost-of-power-in-large-scale-data-centers/