(Jog Falls, Jog, India)

Algorithmic Challenges in Building Efficient Data Center/ Cloud Infrastructure

Janardhan Kulkarni, MSR Redmond.

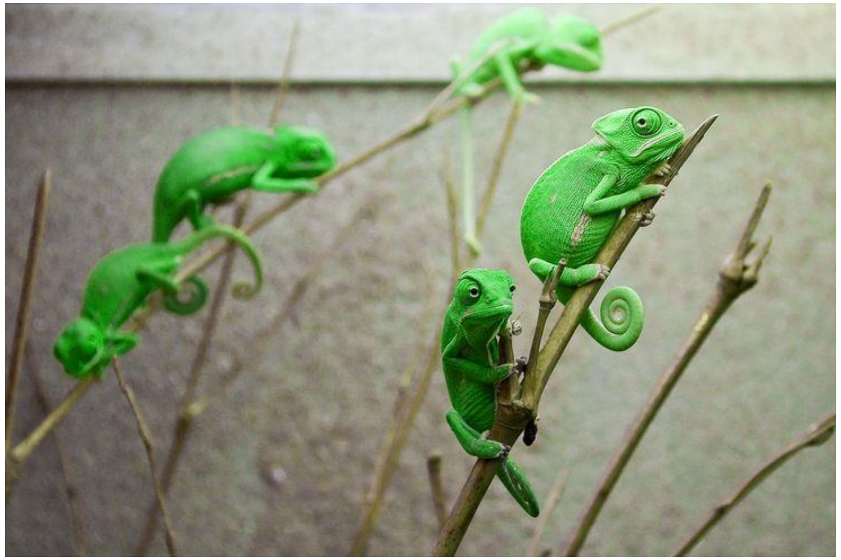
1. Minimum Birkhoff-von Neumann Decompositions
   K., Lee, Singh. IPCO 2017

2. ProjecTor: Agile Reconfigurable Data Center Interconnect
   Ghobadi, Mahajan, Phanishayee, Devanur, K., Ranade, Blanche, Rastegarfar, Glick, Kilper. SIGCOMM'16.

3. Truth and Regret in Online Scheduling
   Chawla, Devanur, K., Niazadeh. EC 2017

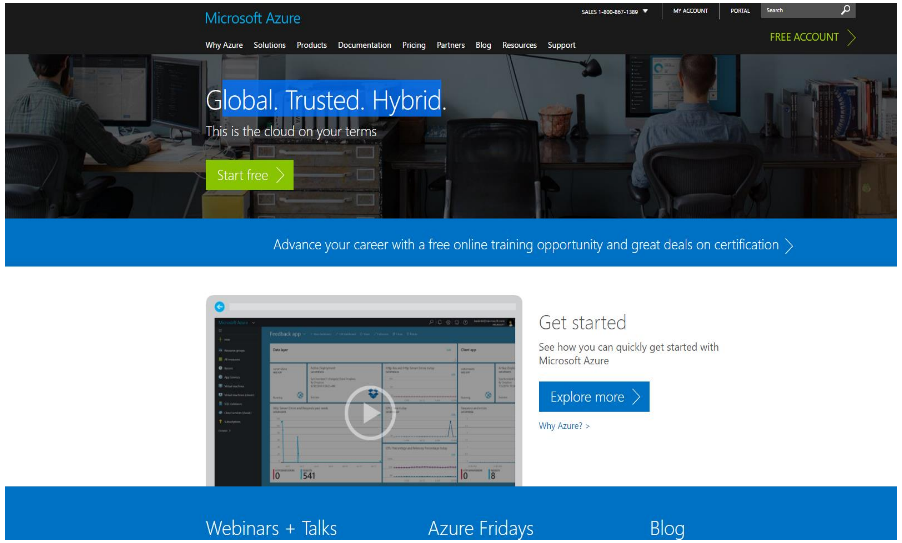
Two Problems in Resource Allocation

Problem 1: Matching Decomposition



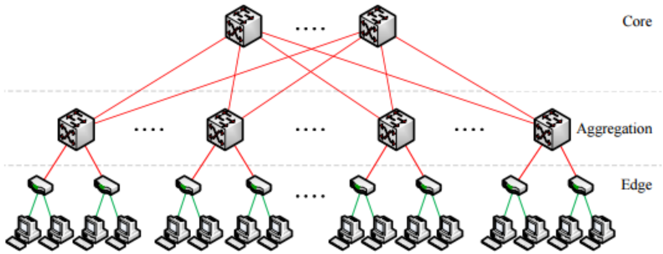
Reconfigurable Data Centers/ SDNs

Problem 2: Pricing and Scheduling VMs

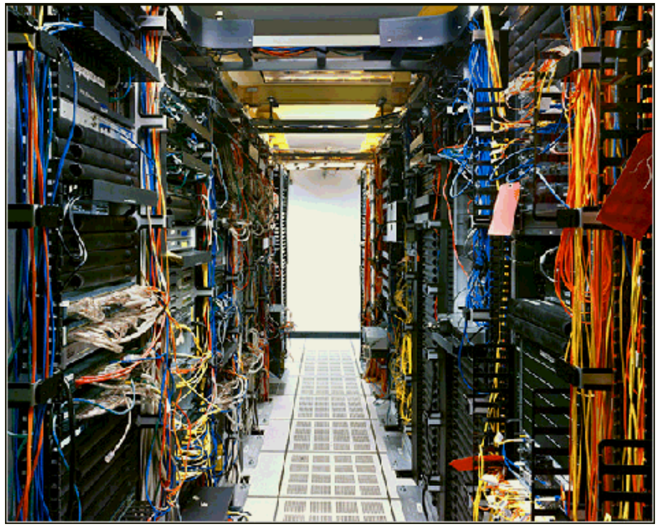


Cloud Services such as Azure

Drawbacks of the Traditional Interconnect



(Al-Fares et al SIGCOMM 08)

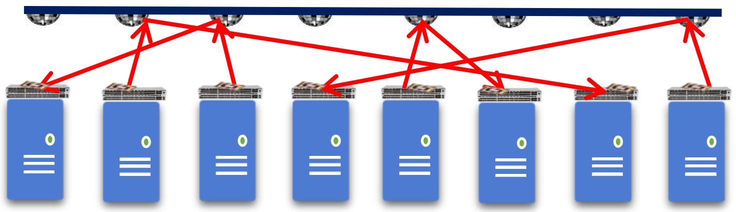


The designers must decide in advance how much capacity to provision between top-of-rack (ToR) switches.

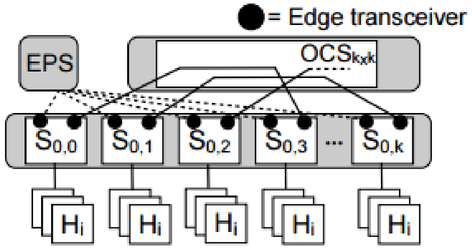
- **Full interconnect is expensive**
- **Limits application performance when demand between two ToRs exceeds capacity**

Reconfigurable Topologies

Change the topology based on traffic!

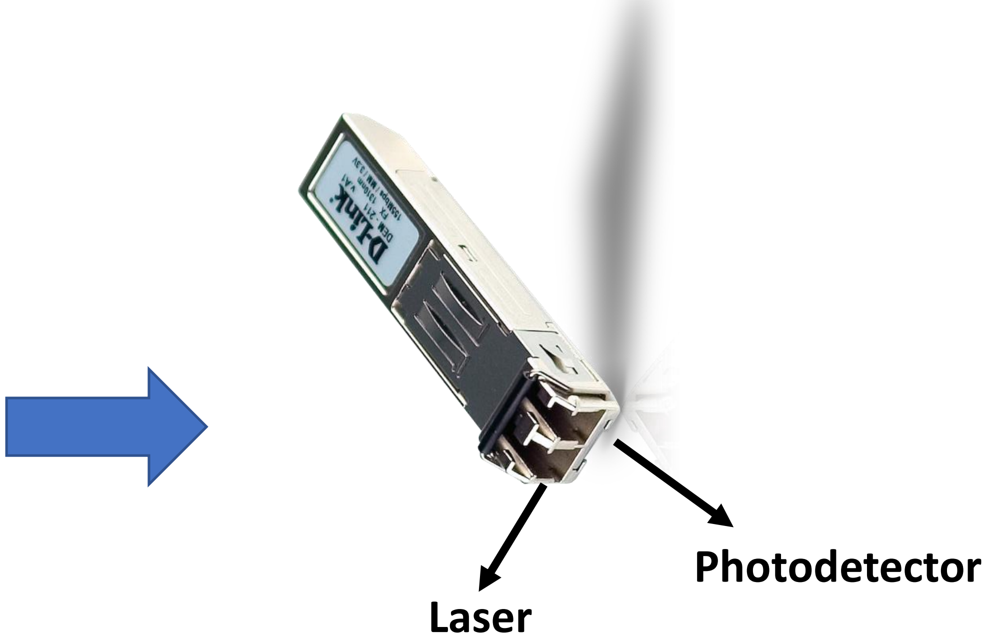


ProjecTor, MSR. (SIGCOMM'16)



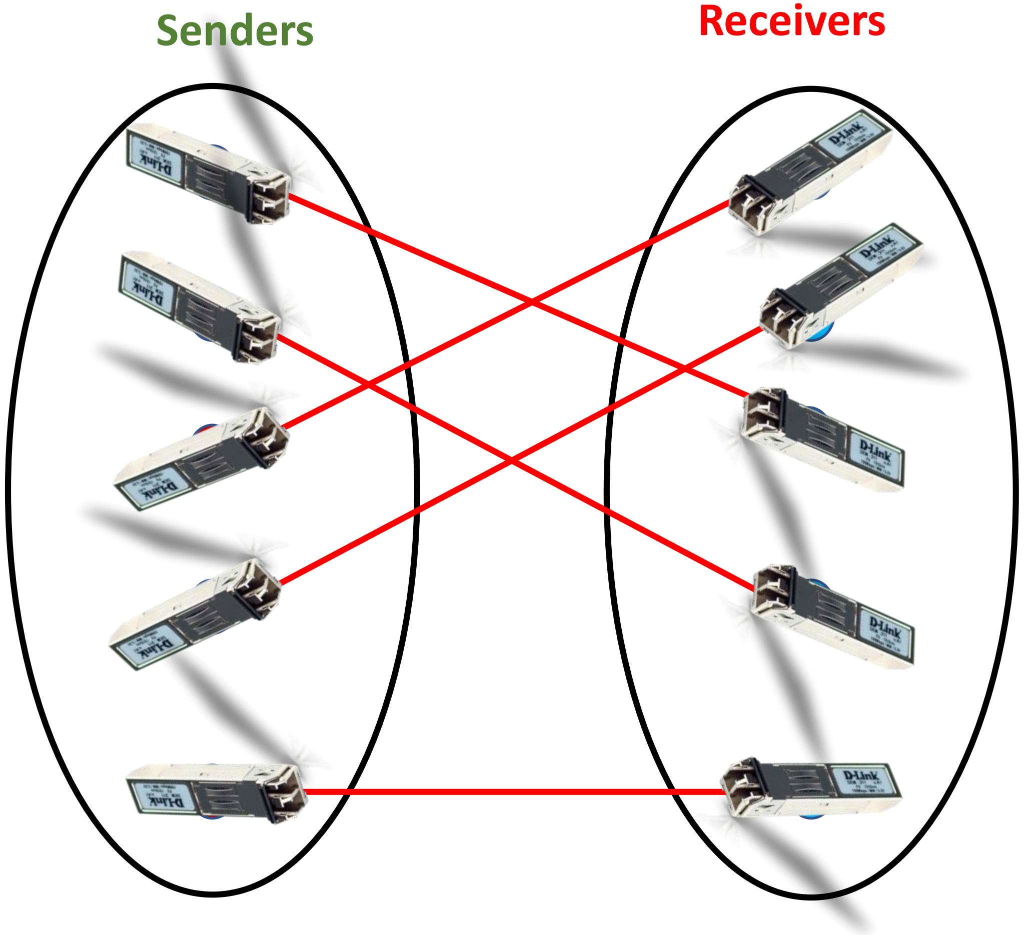
(b) A Hybrid network topology

Mordia, Google. (SIGCOMM'13)



Reconfigurable Topologies

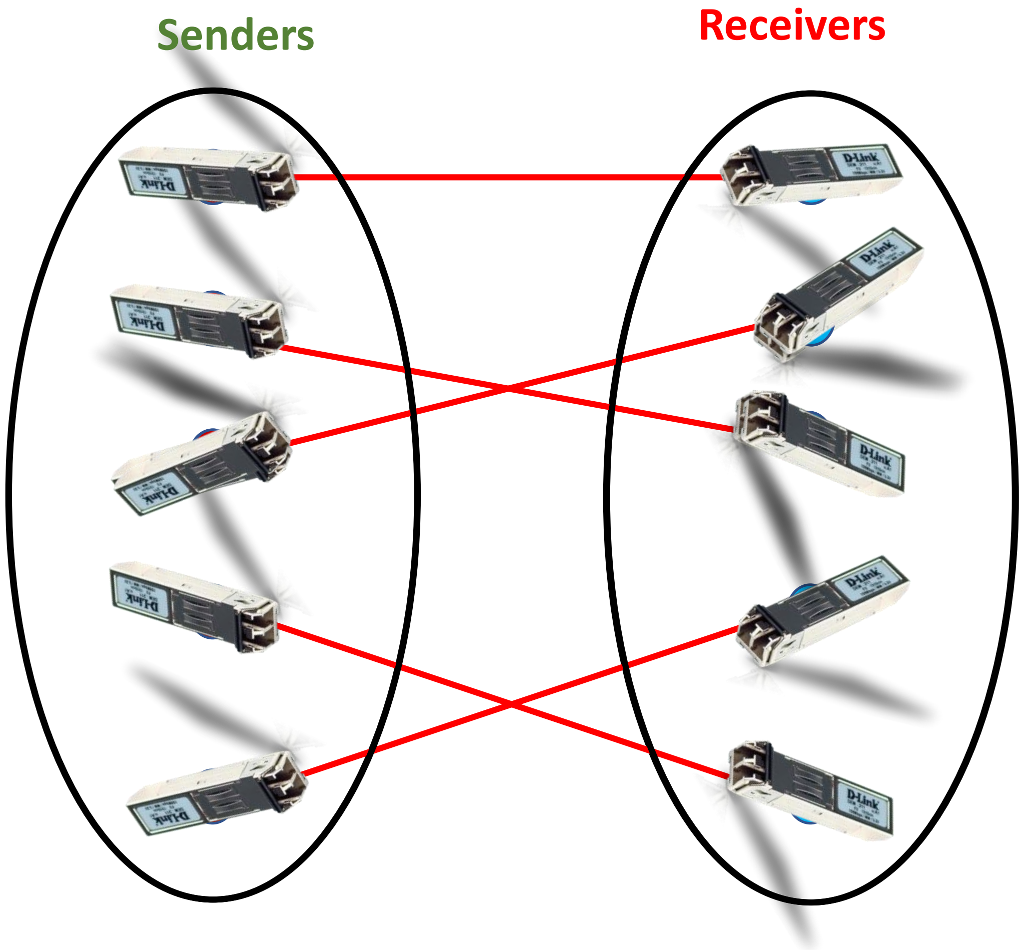
$$
\begin{array}{cc}
 & \text{Receivers} \\
\text{Senders} & \begin{pmatrix}
0 & 60 & 30 & 0 & 90 \\
70 & 0 & 0 & 0 & 50 \\
0 & 20 & 0 & 0 & 100 \\
0 & 30 & 0 & 0 & 40 \\
0 & 20 & 0 & 90 & 0
\end{pmatrix}
\end{array}
$$



Matching between senders and receivers

Reconfigurable Topologies

$$\begin{array}{cc} & \text{Receivers} \\ \text{Senders} & \begin{pmatrix} 0 & 0 & 100 & 0 & 90 \\ 0 & 10 & 60 & 0 & 50 \\ 0 & 20 & 10 & 10 & 0 \\ 0 & 0 & 70 & 50 & 0 \\ 0 & 20 & 0 & 0 & 0 \end{pmatrix} \end{array}$$



Matching between senders and receivers

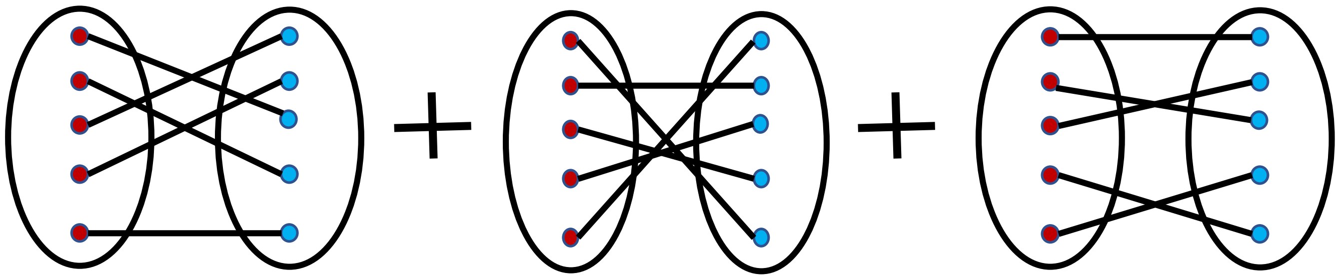
Matching Decomposition

QUESTION

Given a traffic matrix, find an efficient way route the traffic.

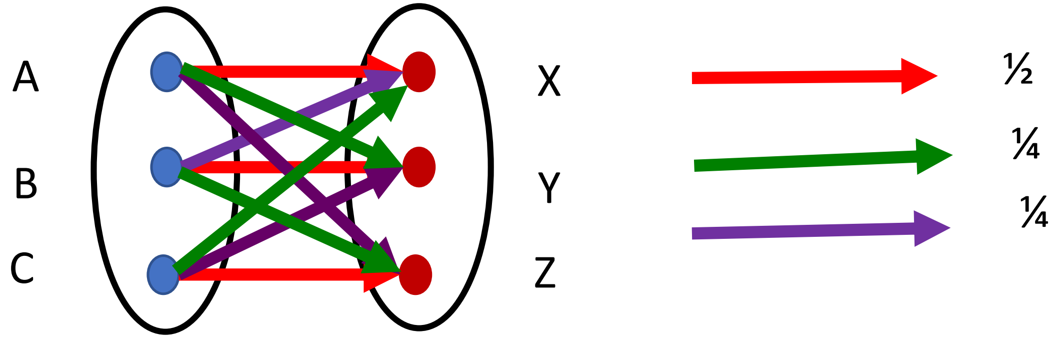
Traffic Matrix | **A sequence of matchings between senders and receivers**

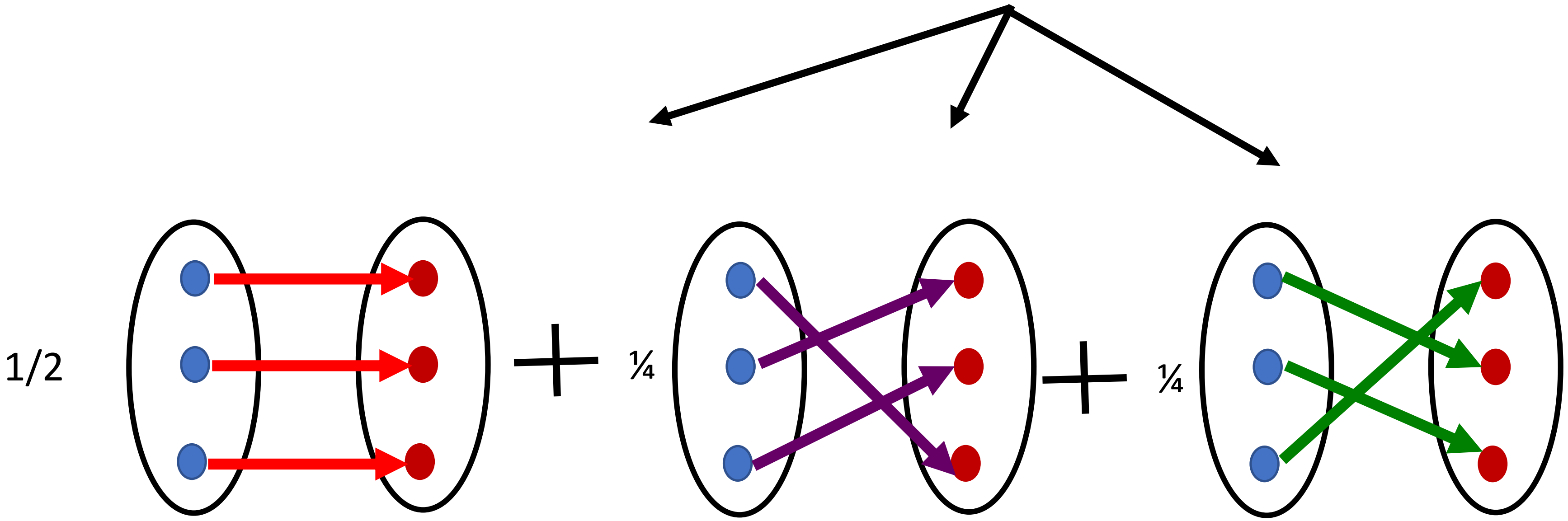
$$\begin{pmatrix} 10 & 20 & 0 & 0 & 5 \\ 0 & 30 & 0 & 10 & 0 \\ 0 & 0 & 90 & 0 & 0 \\ 70 & 20 & 60 & 40 & 10 \end{pmatrix} =$$



Example

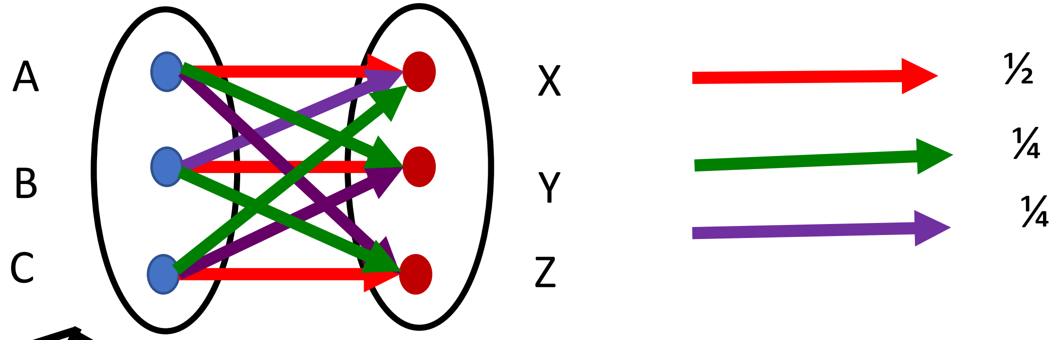
$$
\begin{array}{c|ccc}
 & X & Y & Z \\
\hline
A & \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\
B & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\
C & \frac{1}{4} & \frac{1}{4} & \frac{1}{2}
\end{array}
\quad =
$$

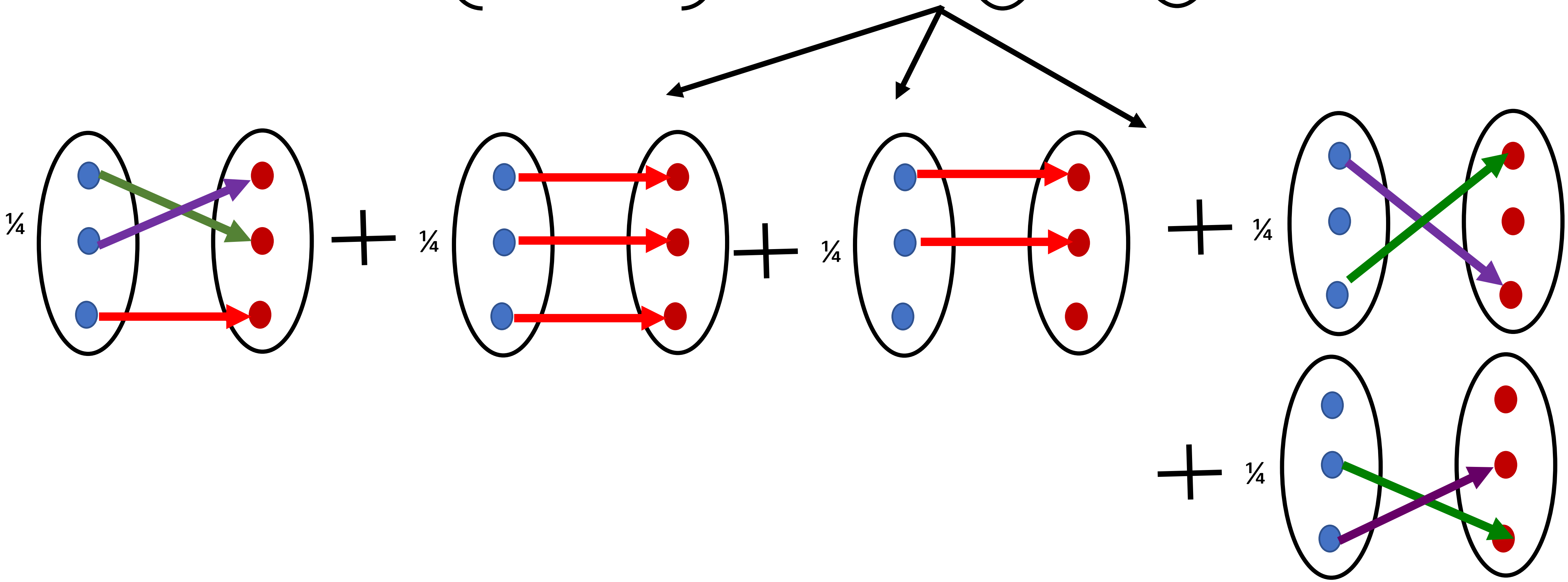




Example

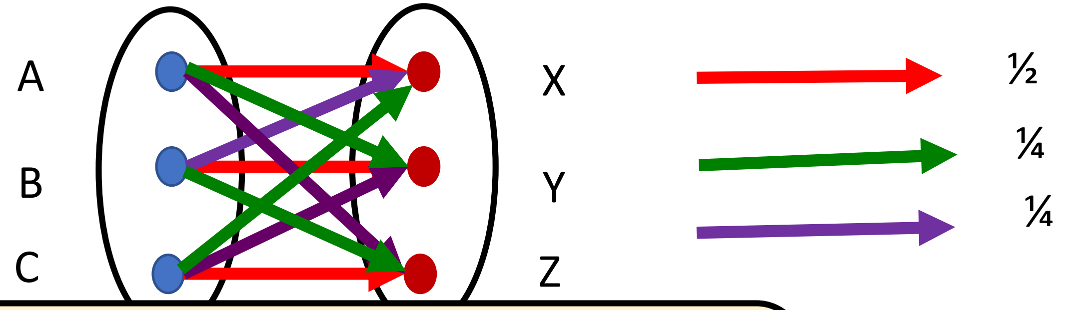
$$\begin{array}{c|ccc} & X & Y & Z \\ \hline A & & & \\ B & & & \\ C & & & \end{array}\quad \begin{array}{c} A \\ B \\ C \end{array}\begin{pmatrix} \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \end{pmatrix} =$$



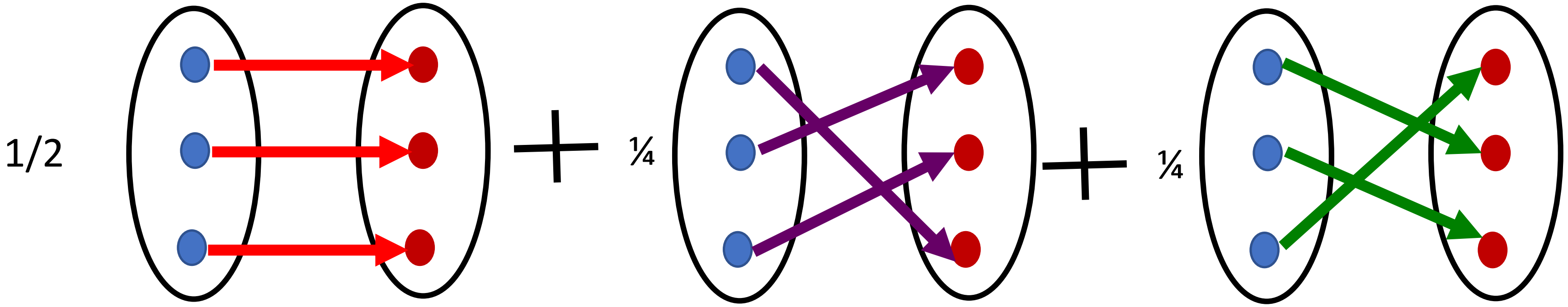


Example

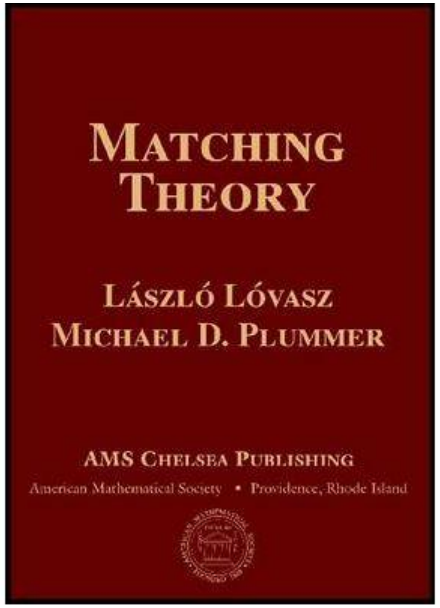
$$\begin{array}{c|ccc} & X & Y & Z \\ \hline A & \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \\ B & \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ C & \frac{1}{4} & \frac{1}{4} & \frac{1}{2} \end{array} =$$



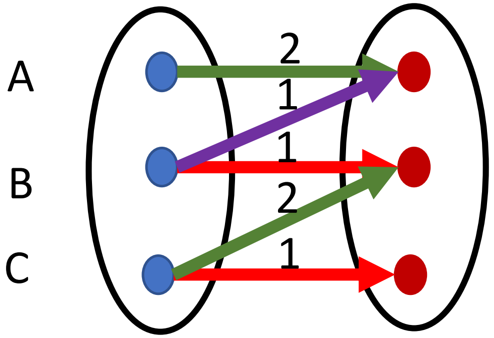
QUESTION: How to decompose a given traffic matrix using smallest number of matchings?

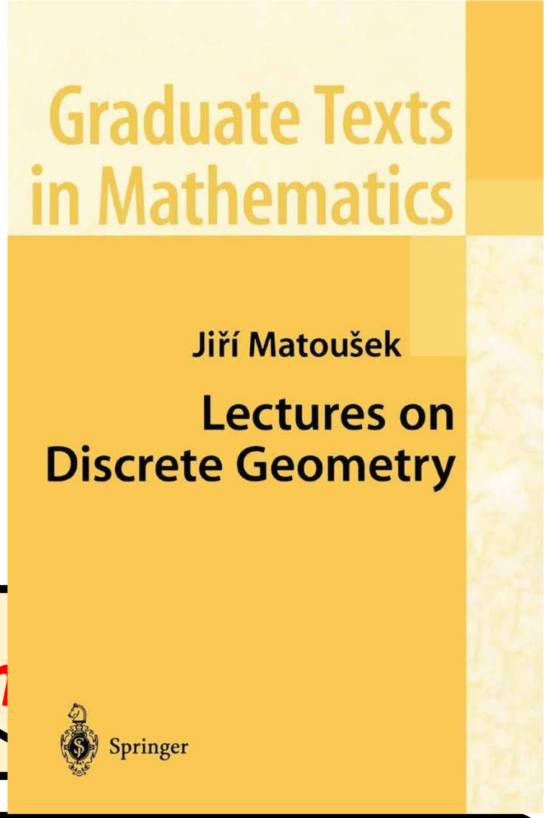


Example

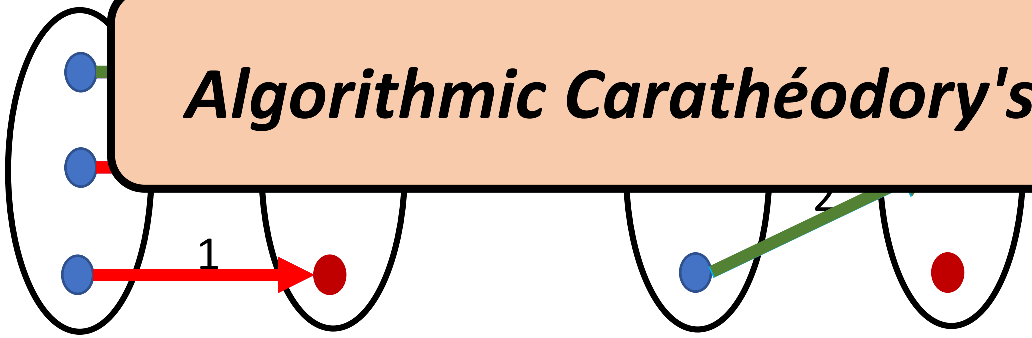


$$\begin{array}{cc} & \begin{array}{ccc} X & Y & Z \end{array} \\ \begin{array}{c} A \\ B \\ C \end{array} & \begin{pmatrix} 2 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 2 & 1 \end{pmatrix} \end{array} =$$





Birkhoff-von Neumann Theorem



Algorithmic Carathéodory's Theorem

Minimum Birkhoff-von Neumann Decompositions

GOAL: $$M = \lambda_1 P_1 + \lambda_2 P_2 + \ldots \lambda_k P_k$$

THEOREM: (K.- Lee- Singh'17)

There is a logarithmic approximation to the minimum Birkhoff-von Nueman decomposition problem.

- **Solve a linear program.**
- **Do randomized rounding.**
- **Apply Lovasz Local Lemma (LLL) to prove the theorem.**

Minimum Birkhoff-von Neumann Decompositions

GOAL: $$M = \lambda_1 P_1 + \lambda_2 P_2 + \ldots \lambda_k P_k$$

THEOREM: (K.- Lee- Singh'17)

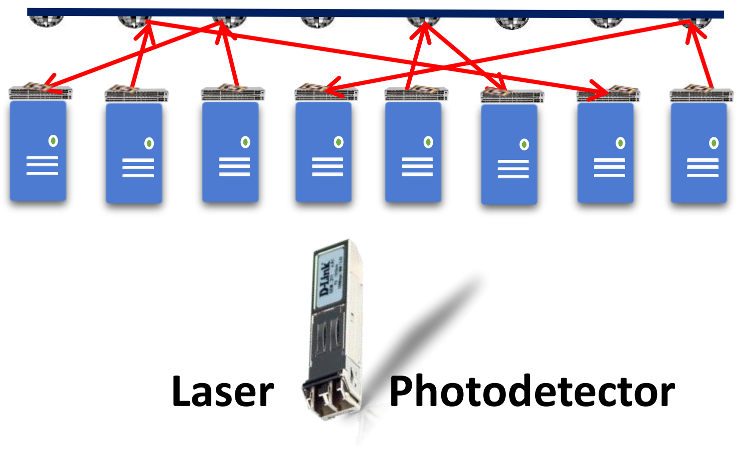
There is a logarithmic approximation to the minimum Birkhoff-von Nueman decomposition problem.

- **Solve a linear program.**
- **Do randomized rounding.**
- **Apply Lovasz Local Lemma (LLL) to prove the theorem.**

BVN Decomposition algorithm can be exponentially bad.

Online + Decentralized Algorithm?

ProjecTor (MSR)

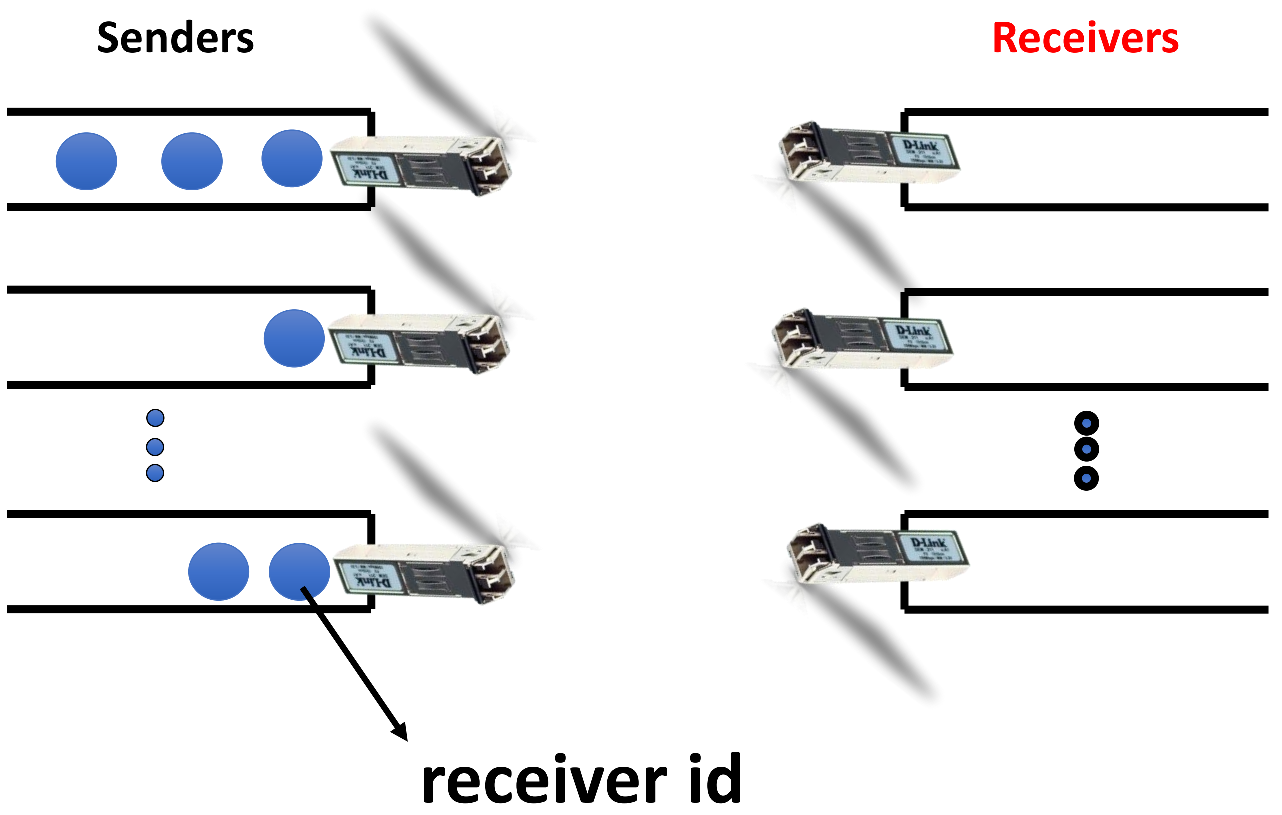


- Online
- Decentralized

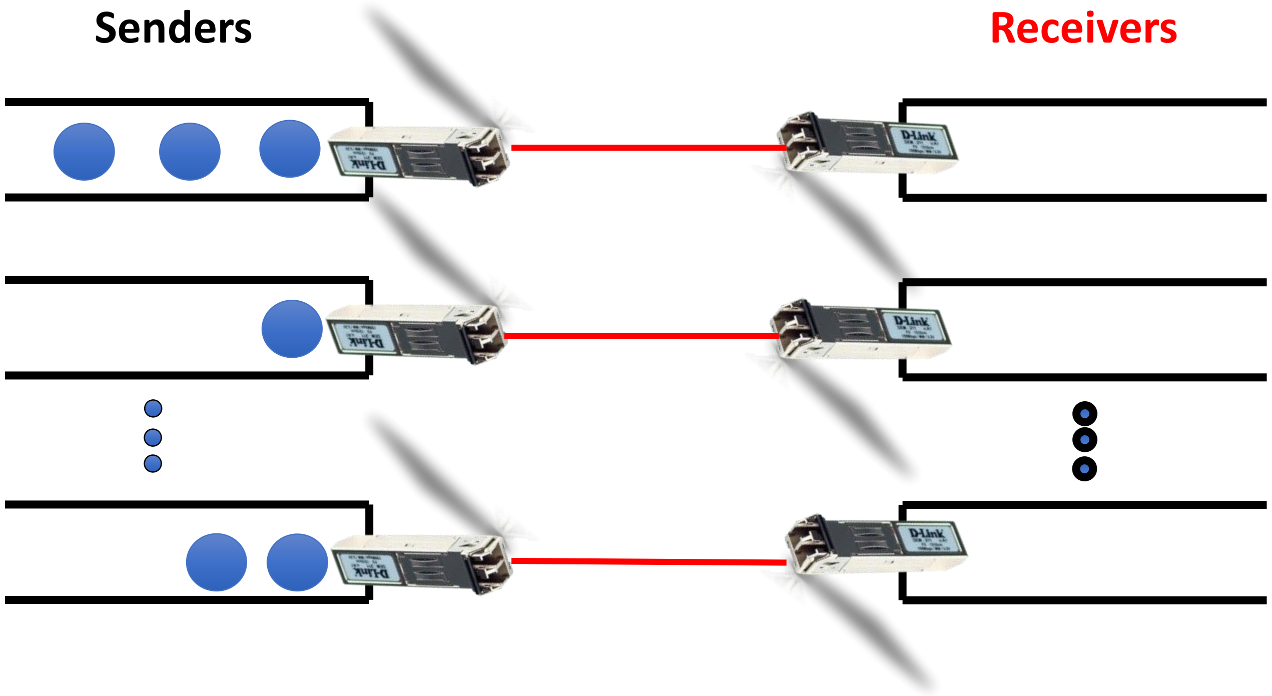
ProjecTor:

Ghobadi, Mahajan, Phanishayee, Devanur, K., Ranade, Blanche, Rastegarfar, Glick, Kilper’16.

Online + Decentralized Algorithm?

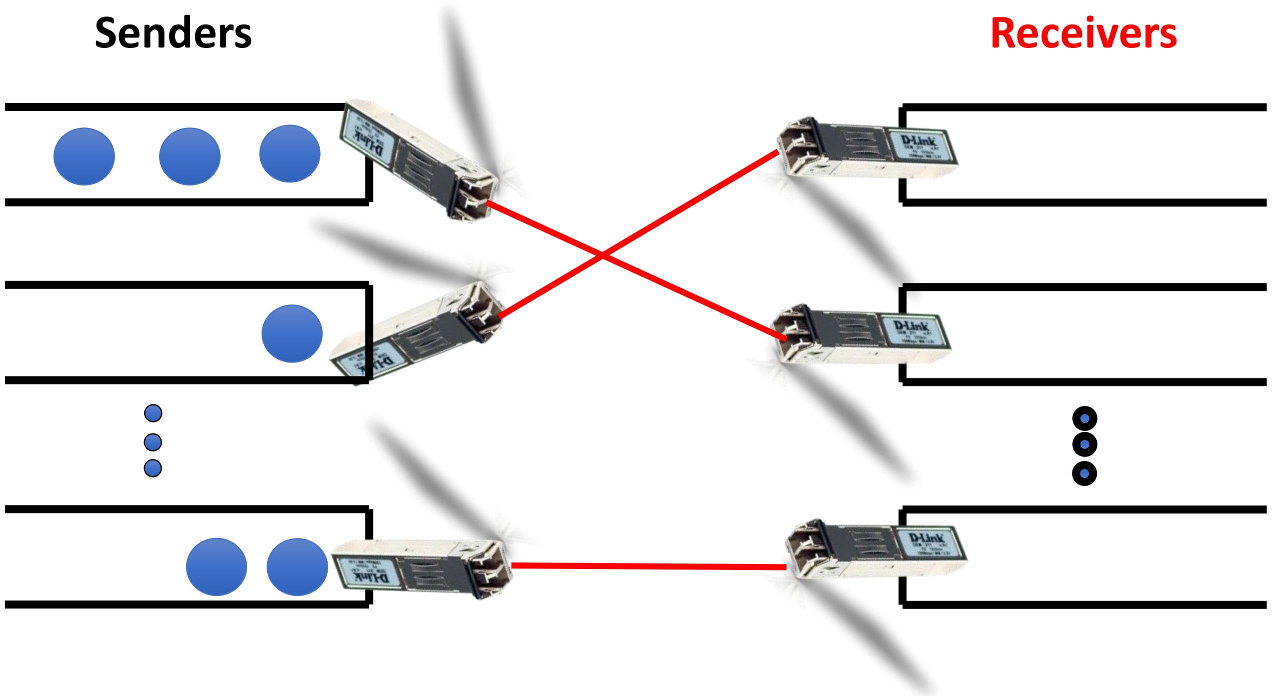


Online + Decentralized Algorithm?



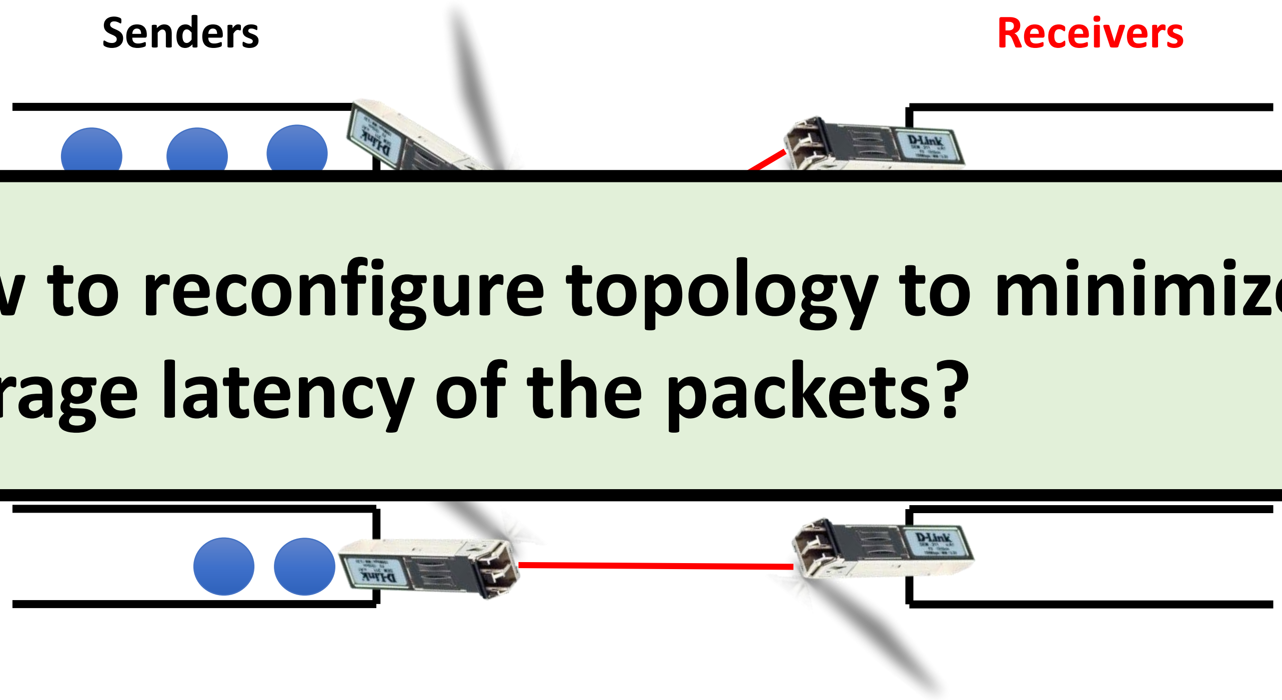
Online algorithm decides on the matching

Online + Decentralized Algorithm?



Online algorithm decides on the matching

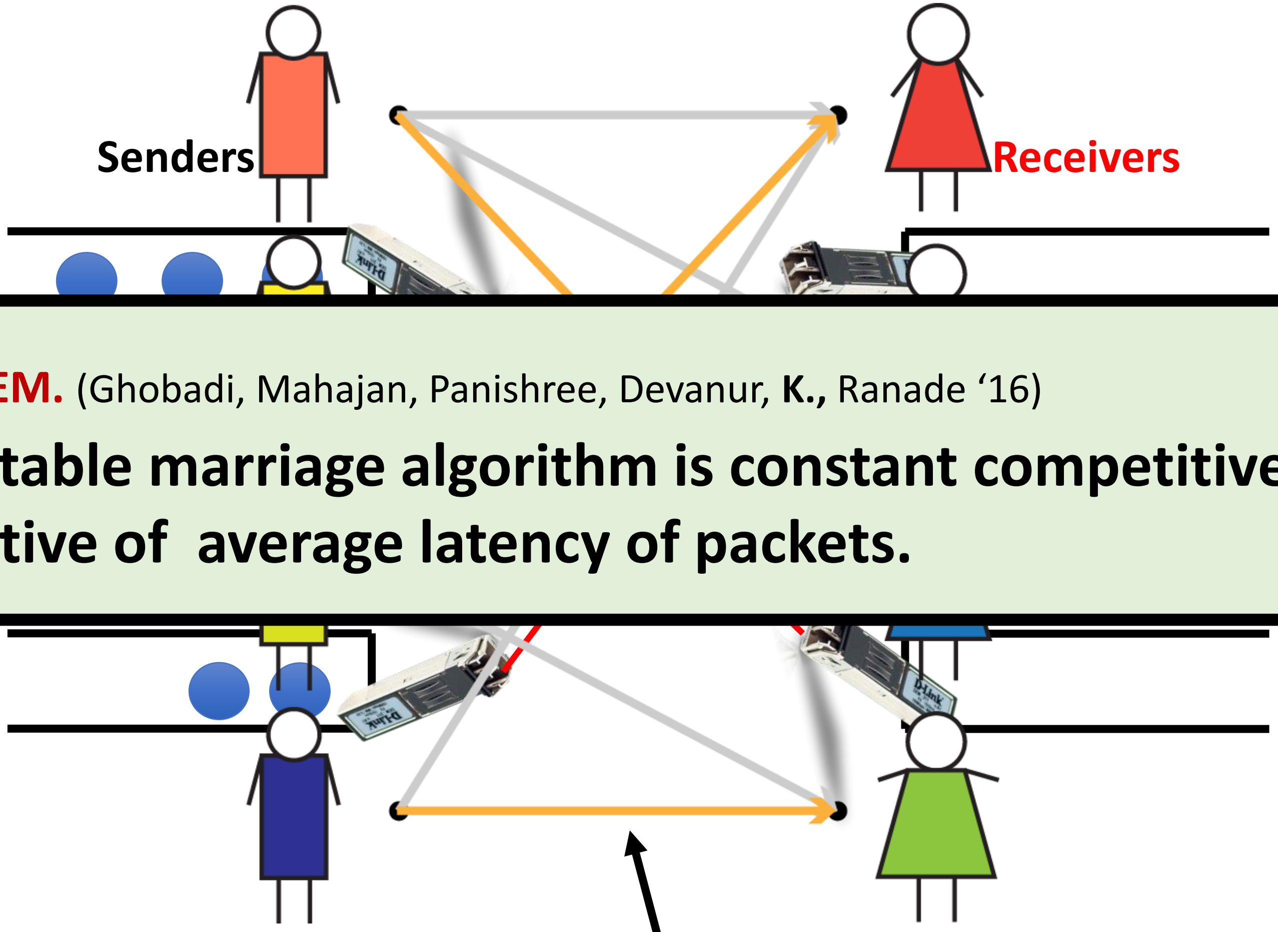
Online + Decentralized Algorithm?



How to reconfigure topology to minimize the average latency of the packets?

Online algorithm decides on the matching

Online + Decentralized Algorithm?



Distribut
Stable M

(Gale, Shapley. 1962.
Nobel Prize 2012)

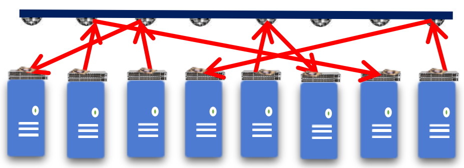
THEOREM. (Ghobadi, Mahajan, Panishree, Devanur, **K.,** Ranade '16)

The Stable marriage algorithm is constant competitive to the objective of average latency of packets.

preference = function of number of packets in queue

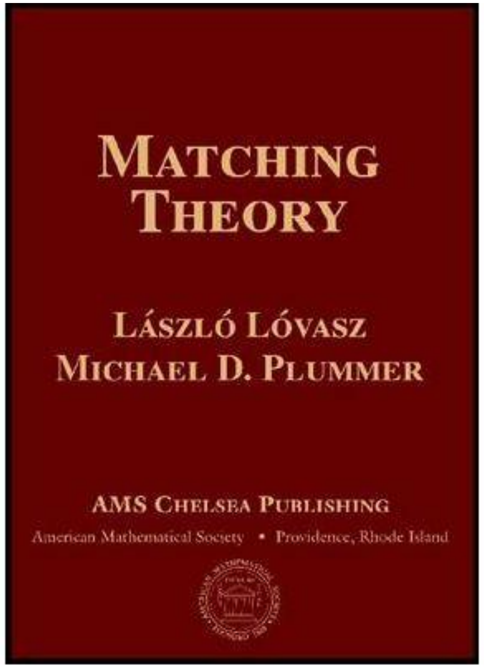
Takeaway

Fundamental Questions in Matching Theory

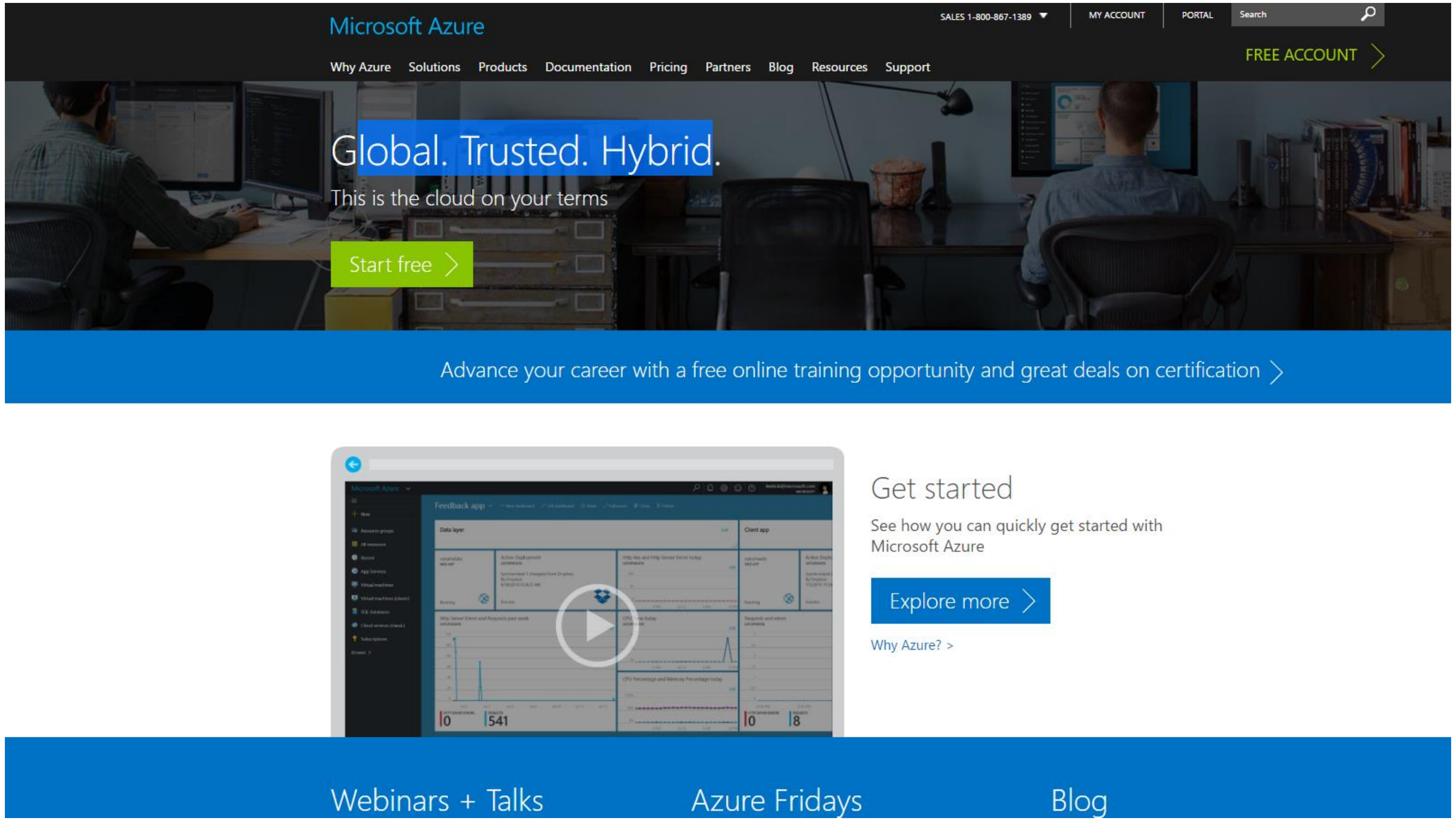


ProjecTor, MSR.

Change the topology based on traffic!

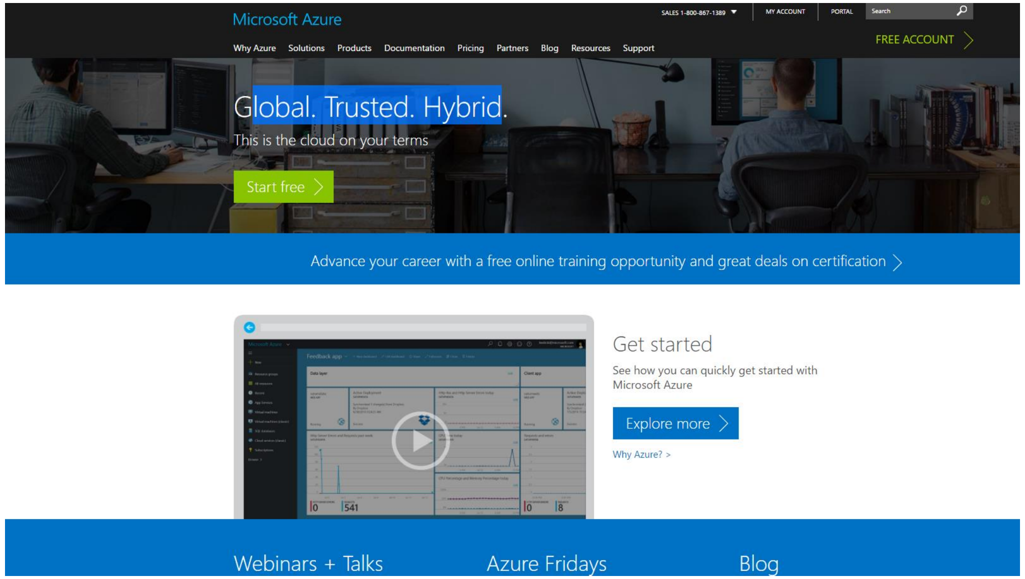


- **Stable marriage algorithm**
- **Algorithmic version of Birkhoff–von Neumann**
- **Algorithmic Carathéodory's Theorem**



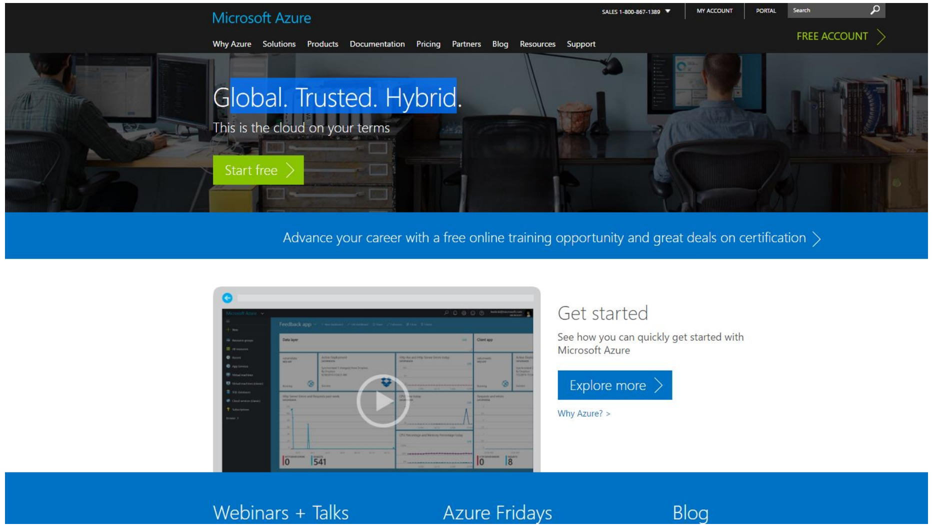
Pricing and Scheduling VMs

Pricing and Scheduling in Azure



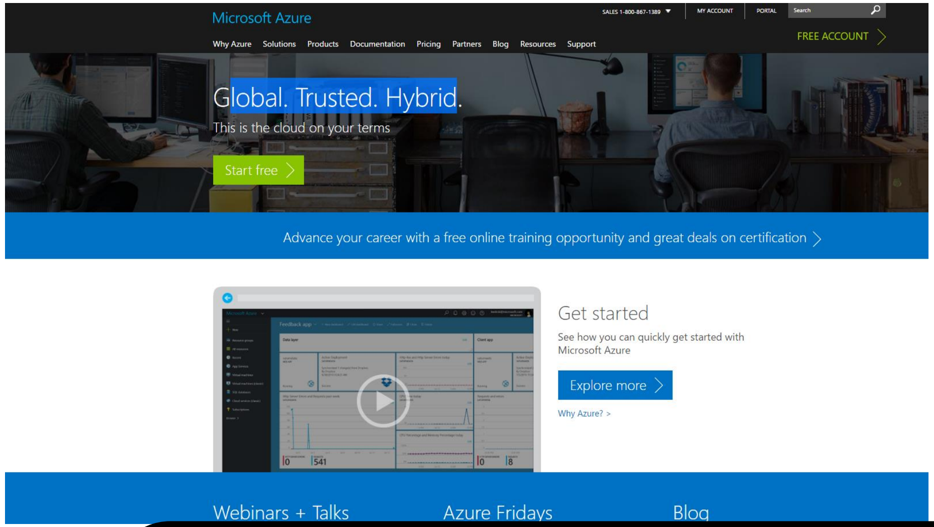
- ➢ **How to price Virtual Machines?**
- ➢ **How to pack/ schedule VMs on a cluster?**

Attempt 1: Modeling the Problem



- A set of jobs arrive online
- Each job has *value*, and *interval of time* where it demands a set of resources.
  - Demands a unit of CPU for some duration ℓ
- Service provider accepts/rejects jobs based on two factors:
  1) Amount of resources available in the system
  2) Value of job

Attempt 1: Modeling the Problem



- A set of jobs arrive online
- Each job has *value*, and *interval of time* where it demands a set of resources.
  - Demands a unit of CPU for some duration ℓ
- Service provider accepts/rejects jobs based on two factors:
  1) Amount of resources available in the system
  2) Value of job

Schedule as many jobs as possible to maximize the total value.

Attempt 1: Modeling the Problem

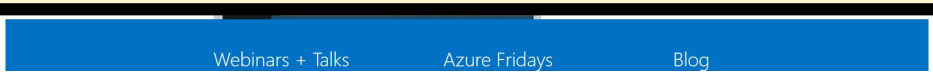
Large literature in online scheduling to maximize throughput.
[ILM'16, JMNY'15, LMNY'13, … CI'98, KSM'94, KS'92…]

Strong lower bounds:
No algorithm can do better than logarithmic factors in the worst case analysis.

rive online

lue, and *interval of time* where it demands
es.
ds a unit of CPU for some duration ℓ

er accepts/rejects jobs based on two factors:
nount of resources available in the system
2) Value of job



Schedule as many jobs as possible to maximize the total value.

Microsoft Azure

SALES 1-800-867-1389 | MY ACCOUNT | PORTAL | Search

Why Azure Solutions Products Documentation Pricing Partners Blog Resources Support

FREE ACCOUNT

Azure Virtual Machines gives you the flexibility of virtualization for a wide range of computing solutions with support for Linux, Windows Server, SQL Server, Oracle, IBM, SAP, and more. Select from a wide variety of virtual machine sizes. Virtual machines are billed on per-minute basis and most include load-balancing and auto-scaling free of charge.

OS/Software: CentOS or Ubuntu Linux

Region: West US 2

Currency: US Dollar ($)

Display pricing by: Hour

Virtual machines categories

– General Purpose

Balanced CPU-to-memory ratio. Ideal for testing and development, small to medium databases, and low to medium traffic web servers.

A0-4 – Basic More information >

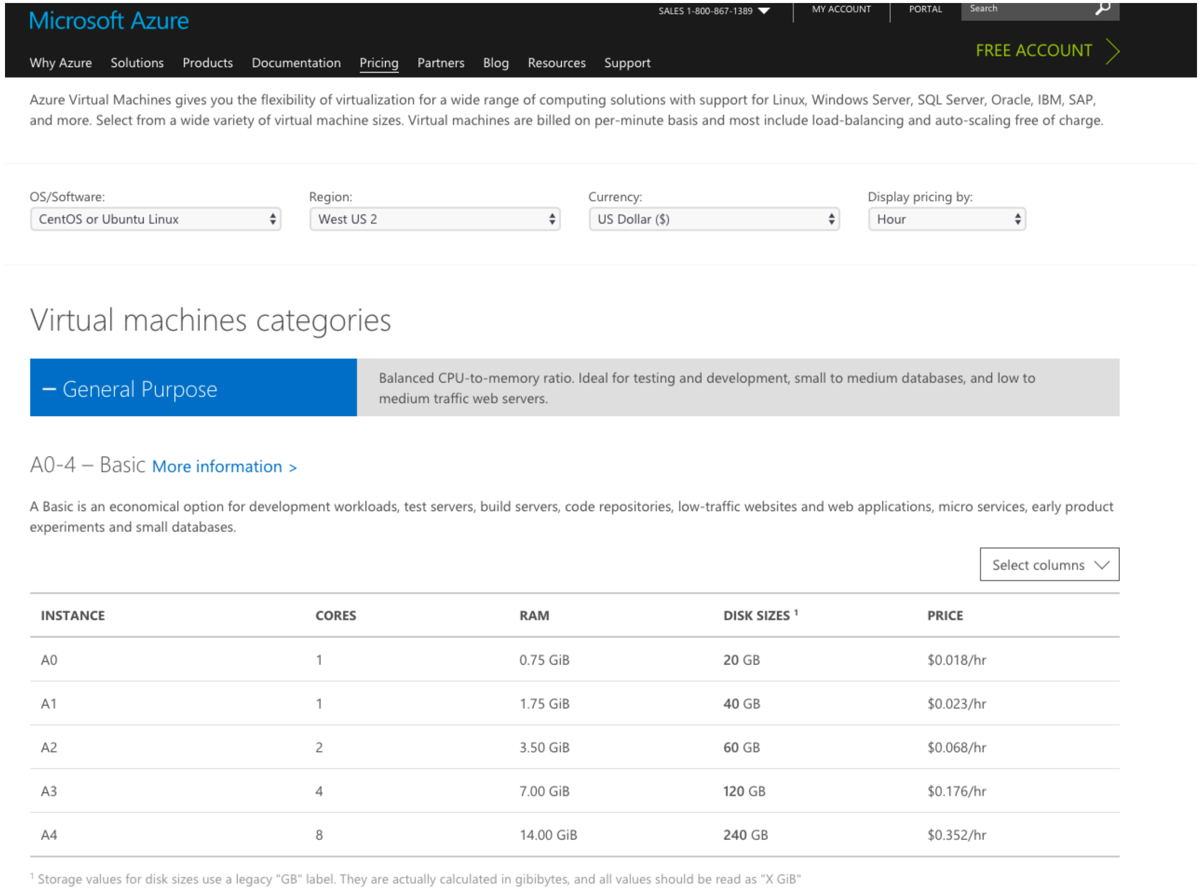
A Basic is an economical option for development workloads, test servers, build servers, code repositories, low-traffic websites and web applications, micro services, early product experiments and small databases.

Select columns

INSTANCE	CORES	RAM	DISK SIZES [1]	PRICE
A0	1	0.75 GiB	**20** GB	$0.018/hr
A1	1	1.75 GiB	**40** GB	$0.023/hr
A2	2	3.50 GiB	**60** GB	$0.068/hr
A3	4	7.00 GiB	**120** GB	$0.176/hr
A4	8	14.00 GiB	**240** GB	$0.352/hr

[1] Storage values for disk sizes use a legacy "GB" label. They are actually calculated in gibibytes, and all values should be read as "X GiB"

Attempt 2: Modeling the Problem



Microsoft Azure

SALES 1-800-867-1389 | MY ACCOUNT | PORTAL | Search

Why Azure Solutions Products Documentation Pricing Partners Blog Resources Support

FREE ACCOUNT

Azure Virtual Machines gives you the flexibility of virtualization for a wide range of computing solutions with support for Linux, Windows Server, SQL Server, Oracle, IBM, SAP, and more. Select from a wide variety of virtual machine sizes. Virtual machines are billed on per-minute basis and most include load-balancing and auto-scaling free of charge.

OS/Software: CentOS or Ubuntu Linux

Region: West US 2

Currency: US Dollar ($)

Display pricing by: Hour

Virtual machines categories

– General Purpose

Balanced CPU-to-memory ratio. Ideal for testing and development, small to medium databases, and low to medium traffic web servers.

A0-4 – Basic More information >

A Basic is an economical option for development workloads, test servers, build servers, code repositories, low-traffic websites and web applications, micro services, early product experiments and small databases.

Select columns

INSTANCE	CORES	RAM	DISK SIZES [1]	PRICE
A0	1	0.75 GiB	20 GB	$0.018/hr
A1	1	1.75 GiB	40 GB	$0.023/hr
A2	2	3.50 GiB	60 GB	$0.068/hr
A3	4	7.00 GiB	120 GB	$0.176/hr
A4	8	14.00 GiB	240 GB	$0.352/hr

[1] Storage values for disk sizes use a legacy "GB" label. They are actually calculated in gibibytes, and all values should be read as "X GiB"

Benchmark

- **Declares a price *p* .**
- **A job that has value per unit length greater than *p* is accepted and scheduled in FIFO order.**
- **Best hindsight price that maximizes the total value of jobs.**

Example

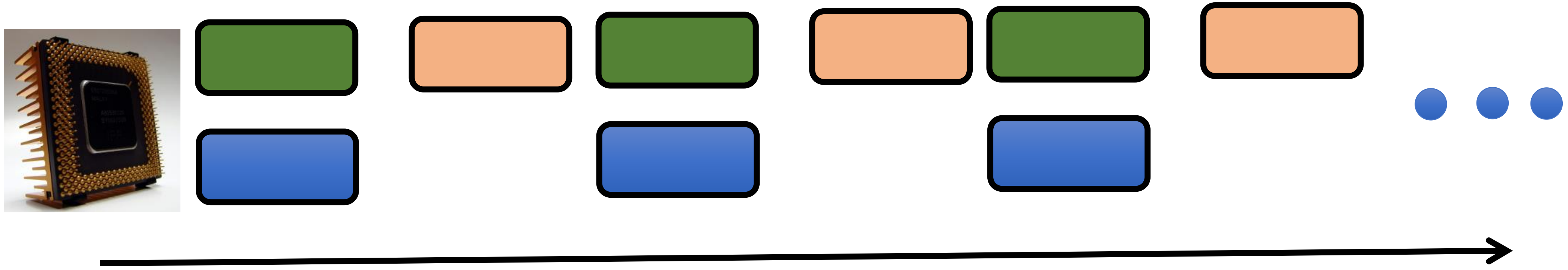
Benchmark: Best hindsight price that maximizes the total value of jobs.
Scheduling Policy: First in First Out (FIFO)

Value = 1, deadline [t, t+1]

Value = 2, deadline [t, t+2]

Value = 2, deadline [t, t+1]

All jobs need one unit of CPU



Example

Benchmark: Best hindsight price that maximizes the total value of jobs.
Scheduling Policy: First in First Out (FIFO)

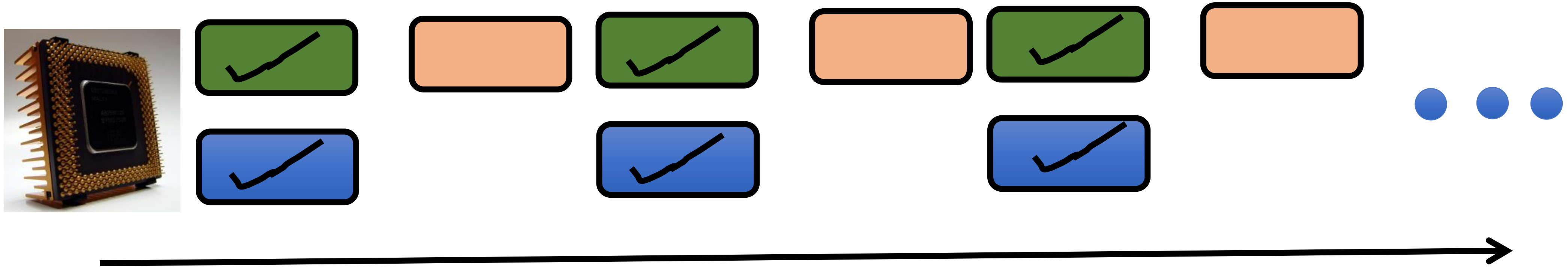
Value = 1, deadline [t, t+1]

Value = 2, deadline [t, t+1]

Value = 2, deadline [t, t+2]

All jobs need one unit of CPU

Price = 1



Total Value (Price = 1) = 3T

Example

Benchmark: Best hindsight price that maximizes the total value of jobs.
Scheduling Policy: First in First Out (FIFO)

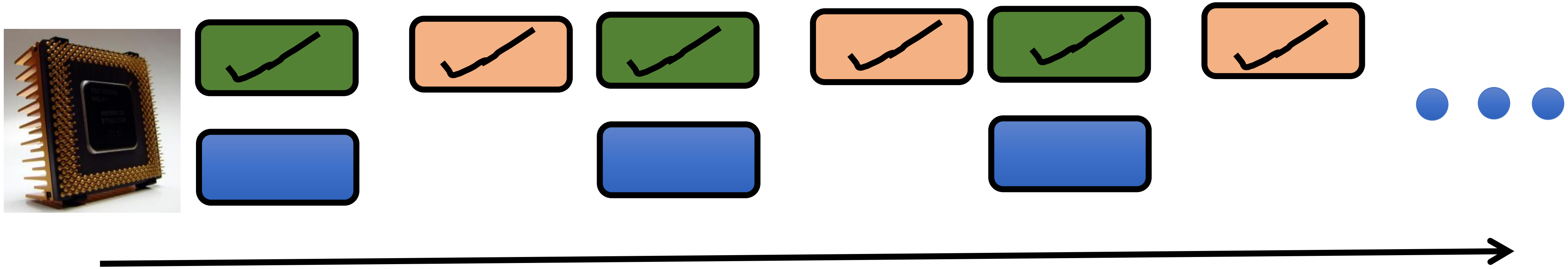
Value = 1, deadline [t, t+1]

Value = 2, deadline [t, t+1]

Value = 2, deadline [t, t+2]

All jobs need one unit of CPU

Price = 2



Total Value (Price = 2) = 4T

Example

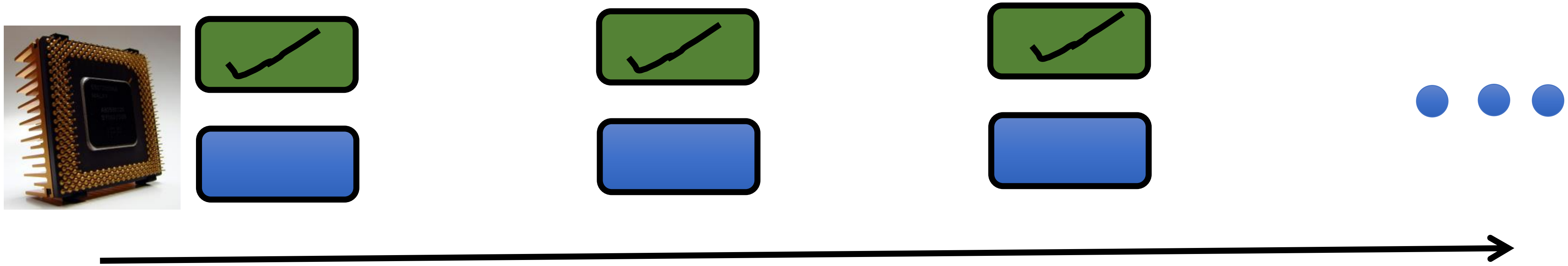
Benchmark: Best hindsight price that maximizes the total value of jobs.
Scheduling Policy: First in First Out (FIFO)

Value = 1, deadline [t, t+1]

Value = 2, deadline [t, t+2]

Both jobs need one unit of CPU

Price = 2



Total Value (Price = 2) = 2T

Example

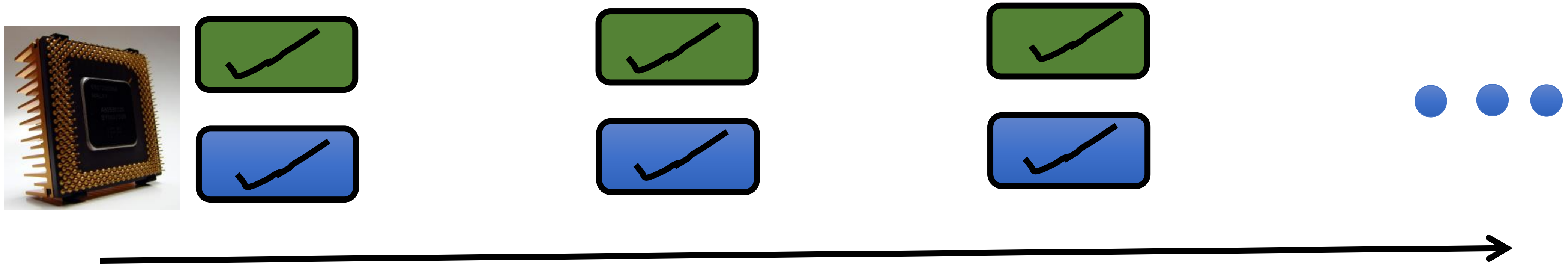
Benchmark: Best hindsight price that maximizes the total value of jobs.
Scheduling Policy: First in First Out (FIFO)

Value = 1, deadline [t, t+1]

Value = 2, deadline [t, t+2]

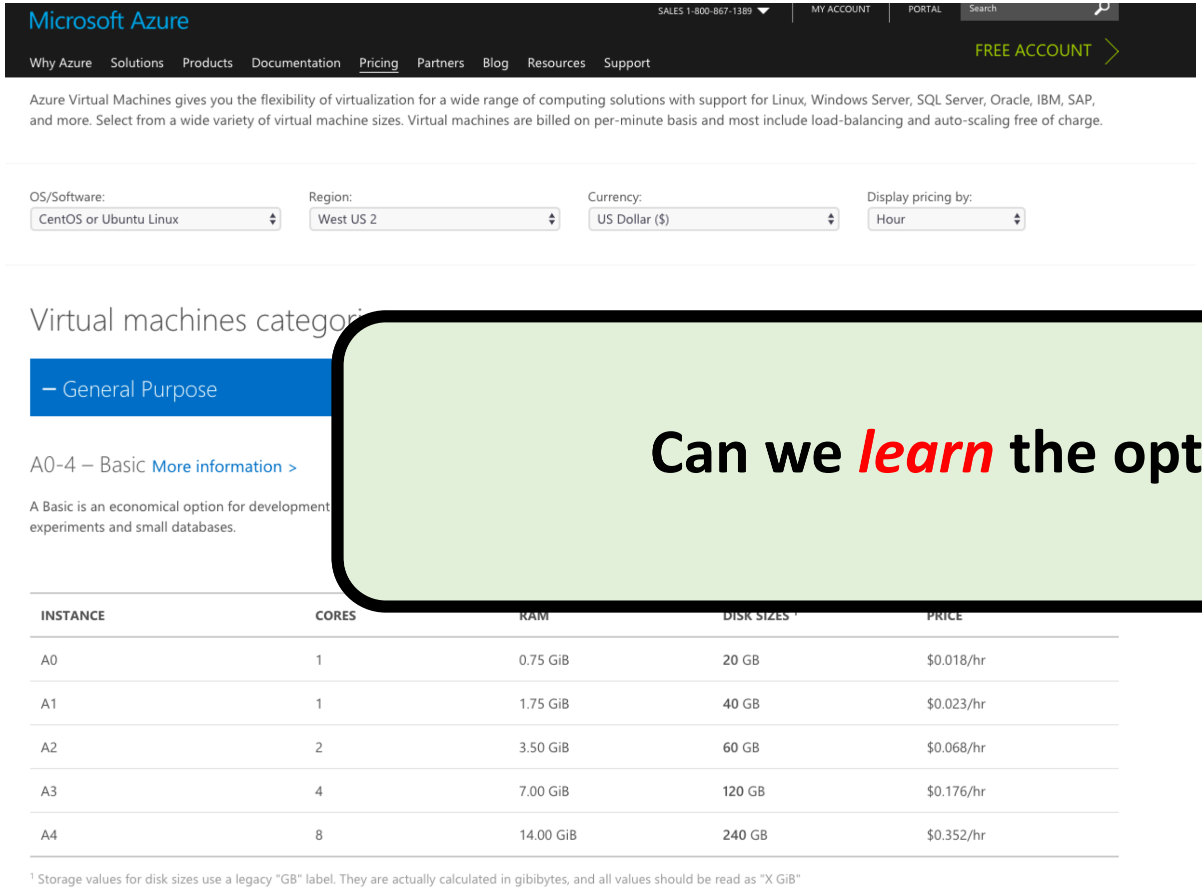
Both jobs need one unit of CPU

Price = 1



Total Value (Price = 1) = 3T

Attempt 2: Modeling the Problem



Microsoft Azure

SALES 1-800-867-1389 | MY ACCOUNT | PORTAL | Search

Why Azure Solutions Products Documentation Pricing Partners Blog Resources Support

FREE ACCOUNT

Azure Virtual Machines gives you the flexibility of virtualization for a wide range of computing solutions with support for Linux, Windows Server, SQL Server, Oracle, IBM, SAP, and more. Select from a wide variety of virtual machine sizes. Virtual machines are billed on per-minute basis and most include load-balancing and auto-scaling free of charge.

OS/Software: CentOS or Ubuntu Linux

Region: West US 2

Currency: US Dollar ($)

Display pricing by: Hour

Virtual machines categori

– General Purpose

A0-4 – Basic More information >

A Basic is an economical option for development
experiments and small databases.

INSTANCE	CORES	RAM	DISK SIZES [1]	PRICE
A0	1	0.75 GiB	20 GB	$0.018/hr
A1	1	1.75 GiB	40 GB	$0.023/hr
A2	2	3.50 GiB	60 GB	$0.068/hr
A3	4	7.00 GiB	120 GB	$0.176/hr
A4	8	14.00 GiB	240 GB	$0.352/hr

[1] Storage values for disk sizes use a legacy "GB" label. They are actually calculated in gibibytes, and all values should be read as "X GiB"

Benchmark: Best hindsight price that

f jobs.

Can we *learn* the optimal price?

Regret Analysis

The online algorithm can change/adapt its price over time.

Benchmark: Best hindsight price that maximizes the total value of jobs.

Regret = Total Value (p^*) - Total Value of ALG

Regret Analysis

The online algorithm can change/adapt its price over time.

Benchmark: Best hindsight price that maximizes the total value of jobs.

$$\frac{\textbf{Regret}}{T} = \frac{\textbf{Total Value (p*)}}{T} - \frac{\textbf{Total Value of ALG}}{T}$$

Good Learning Algorithm: *Average regret approaches zero as time increases.*

Regret Analysis

The online algorithm can change/adapt its price over time.

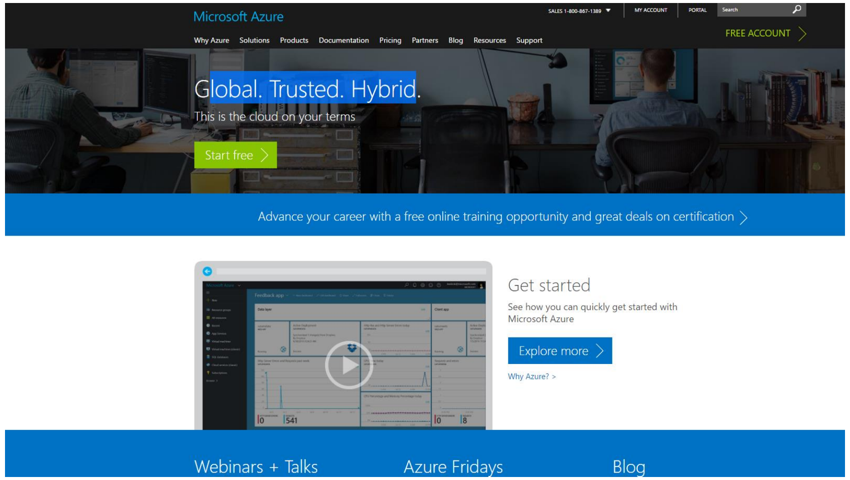
Benchmark: Best hindsight price that maximizes the total value of jobs.

$$\frac{\textbf{Regret}}{T} = \frac{\textbf{Total Value (p*)}}{T} - \frac{\textbf{Total Value of ALG}}{T}$$

Good Learning Algorithm: ***Average regret approaches zero as time increases.***

Chawla et al ' 17: For iid distributions, optimal solution is a pricing algorithm.

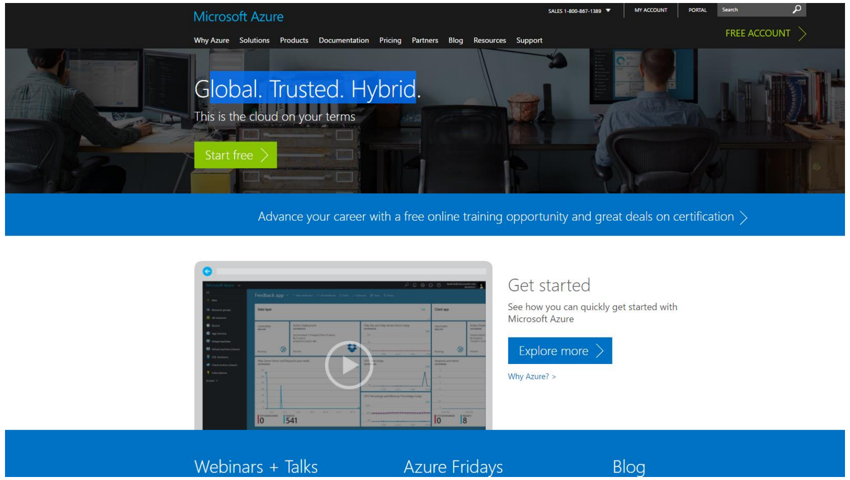
Optimal Learning Algorithm



THEOREM: Chawla-Devanur-K.-Niazadeh'17

There is an online learning algorithm that *achieves optimal* regret for the problem of scheduling to jobs to maximize total value.

Optimal Learning Algorithm



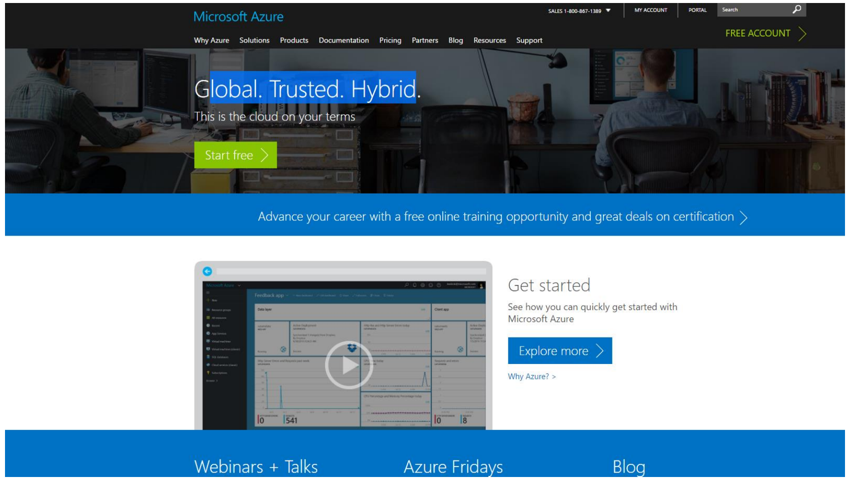
THEOREM: Chawla-Devanur-K.-Niazadeh'17

There is an online learning algorithm that *achieves optimal* regret for the problem of scheduling to jobs to maximize total value.

Truthful!

(Jobs have no incentive to lie about their value, deadlines and arrivals.)

Optimal Learning Algorithm



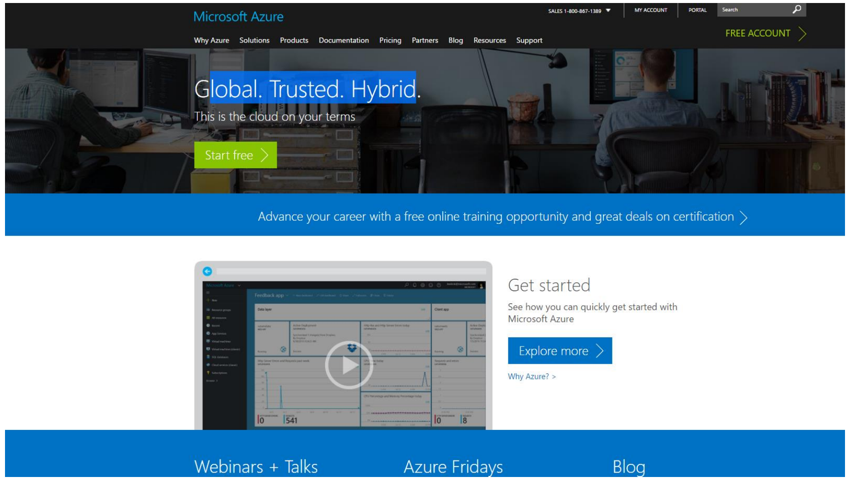
THEOREM: Chawla-Devanur-K.-Niazadeh'17

There is an online learning algorithm that *achieves optimal* regret for the problem of scheduling to jobs to maximize total value.

$O(\sqrt{T})$ **Regret in the case when job lengths are known in advance**

$O(T^{2/3})$ **Regret in the case when job lengths are *not* known in advance**

Optimal Learning Algorithm



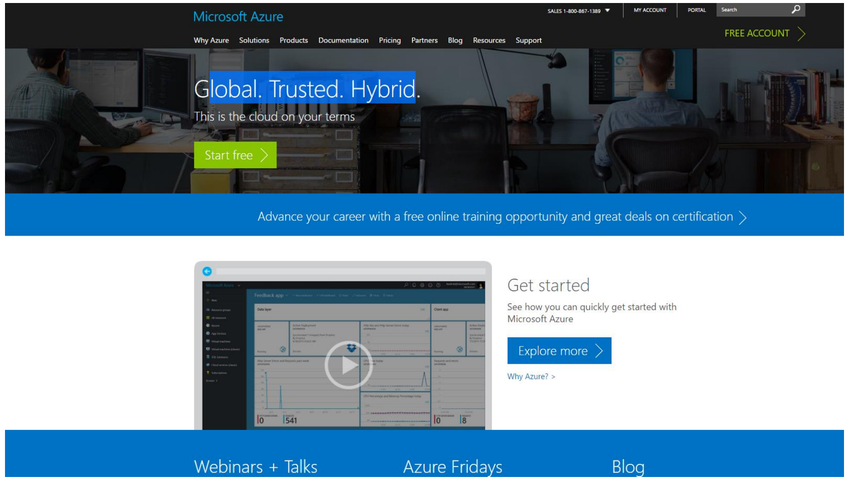
THEOREM: Chawla-Devanur-K.-Niazadeh'17

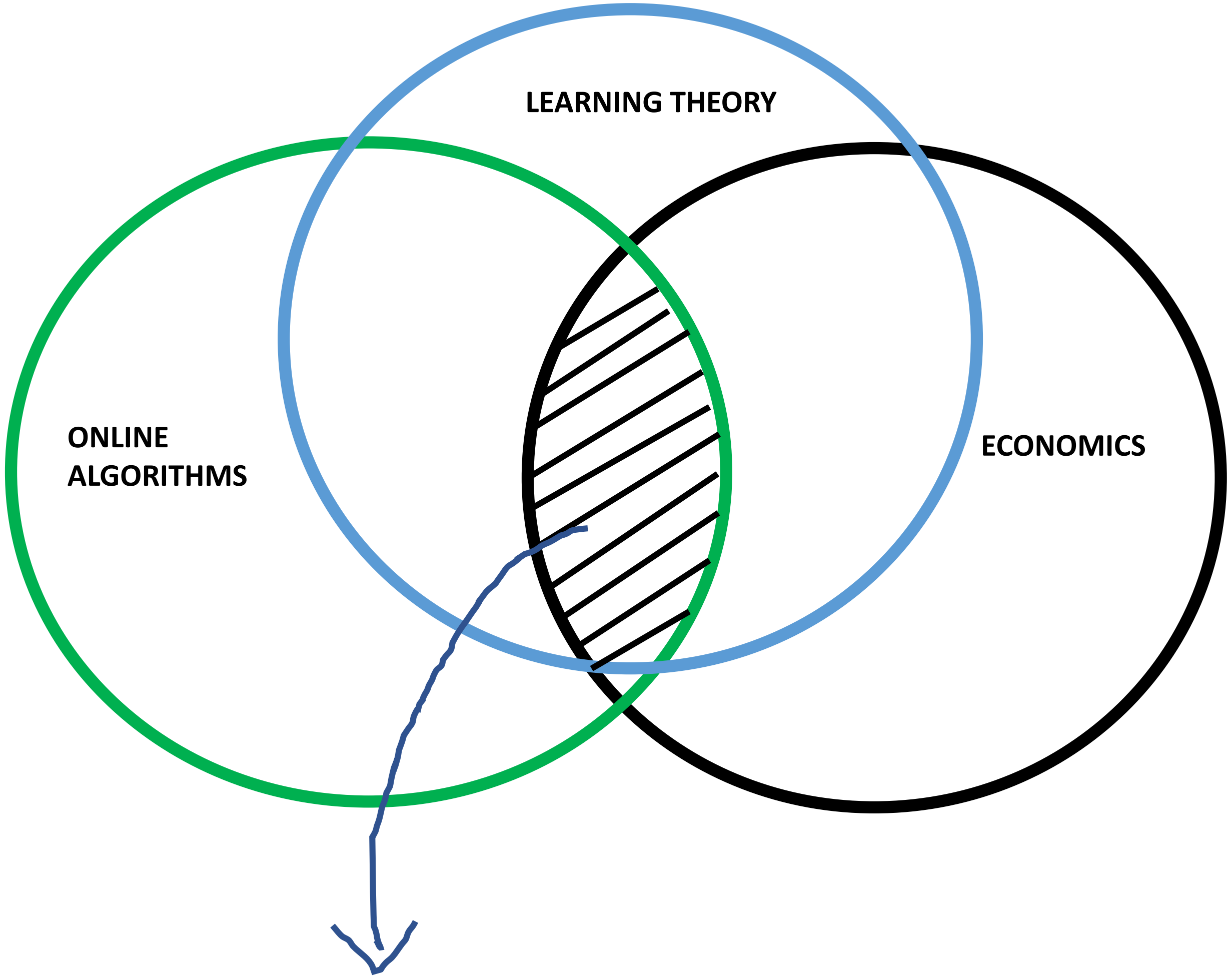
There is an online learning algorithm that ***achieves optimal*** **regret for the problem of scheduling to jobs to maximize total value.**

MAIN IDEA

- ***Think of each price as an expert.***
- ***Scheduling problem has a state.***
- ***Adaptations of algorithms from experts/bandit with with switching cost model.***

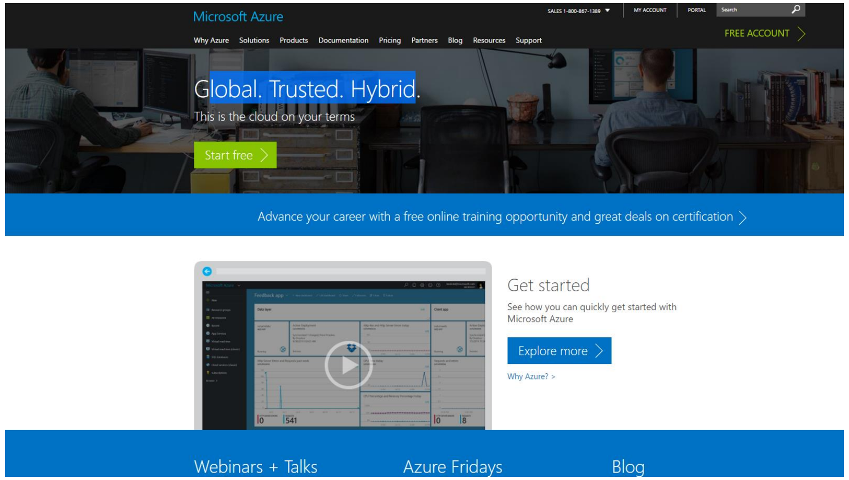
Takeaway

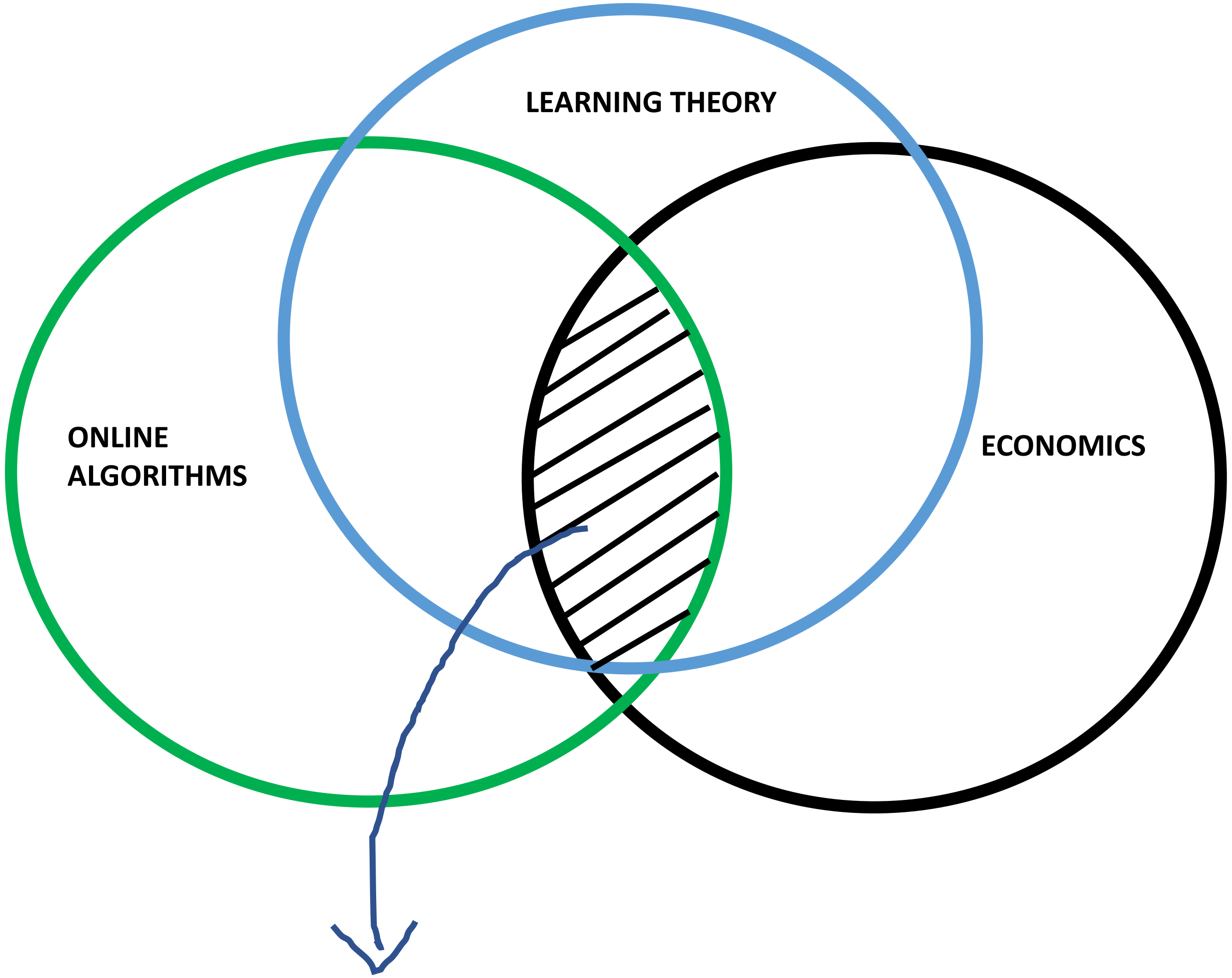




Interesting Scheduling Problems at the Intersection

Takeaway





Interesting Scheduling Problems at the Intersection

Thank You