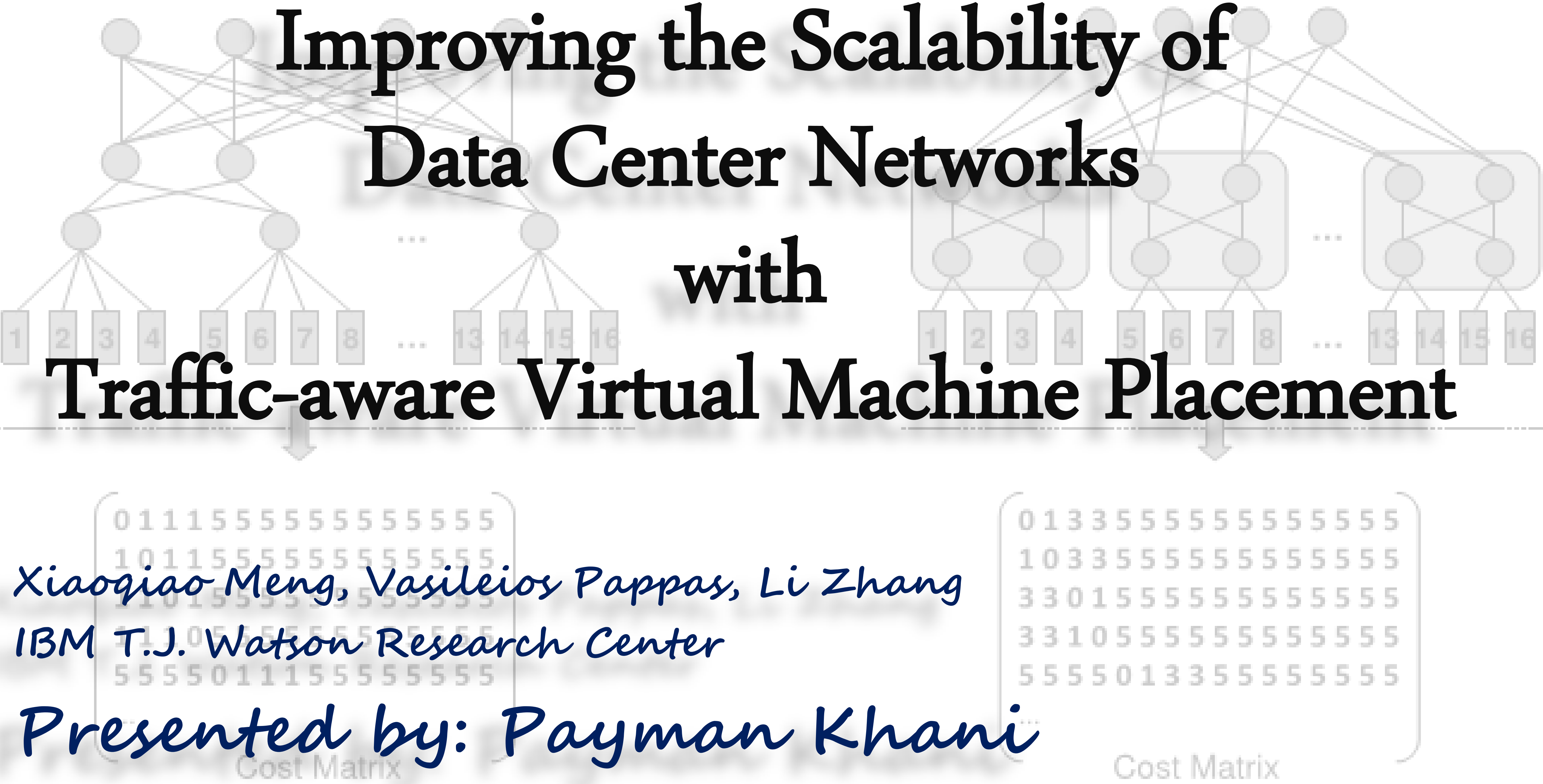
Improving the Scalability of Data Center Networks with Traffic-aware Virtual Machine Placement

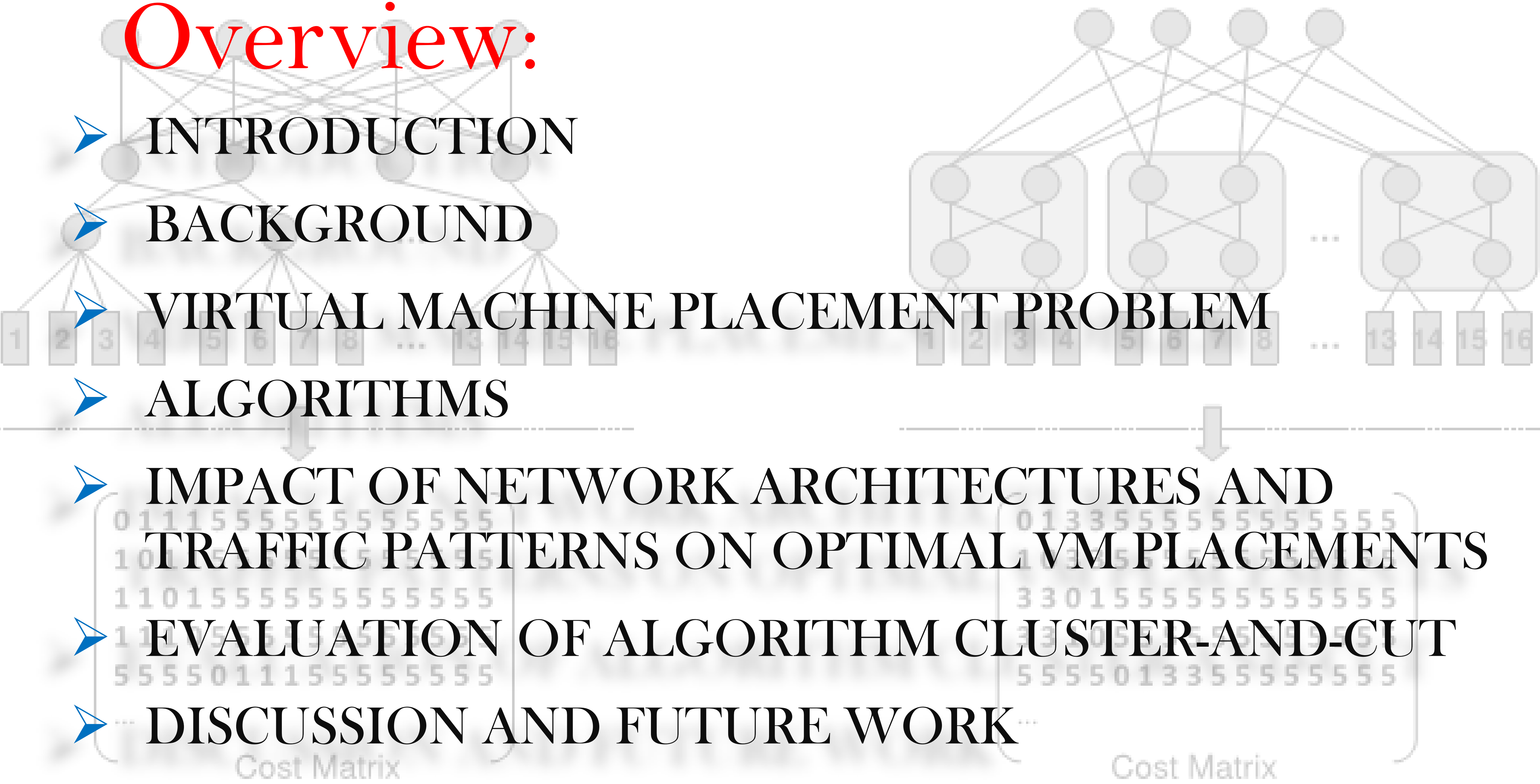
Xiaoqiao Meng, Vasileios Pappas, Li Zhang
IBM T.J. Watson Research Center

Presented by: Payman Khani

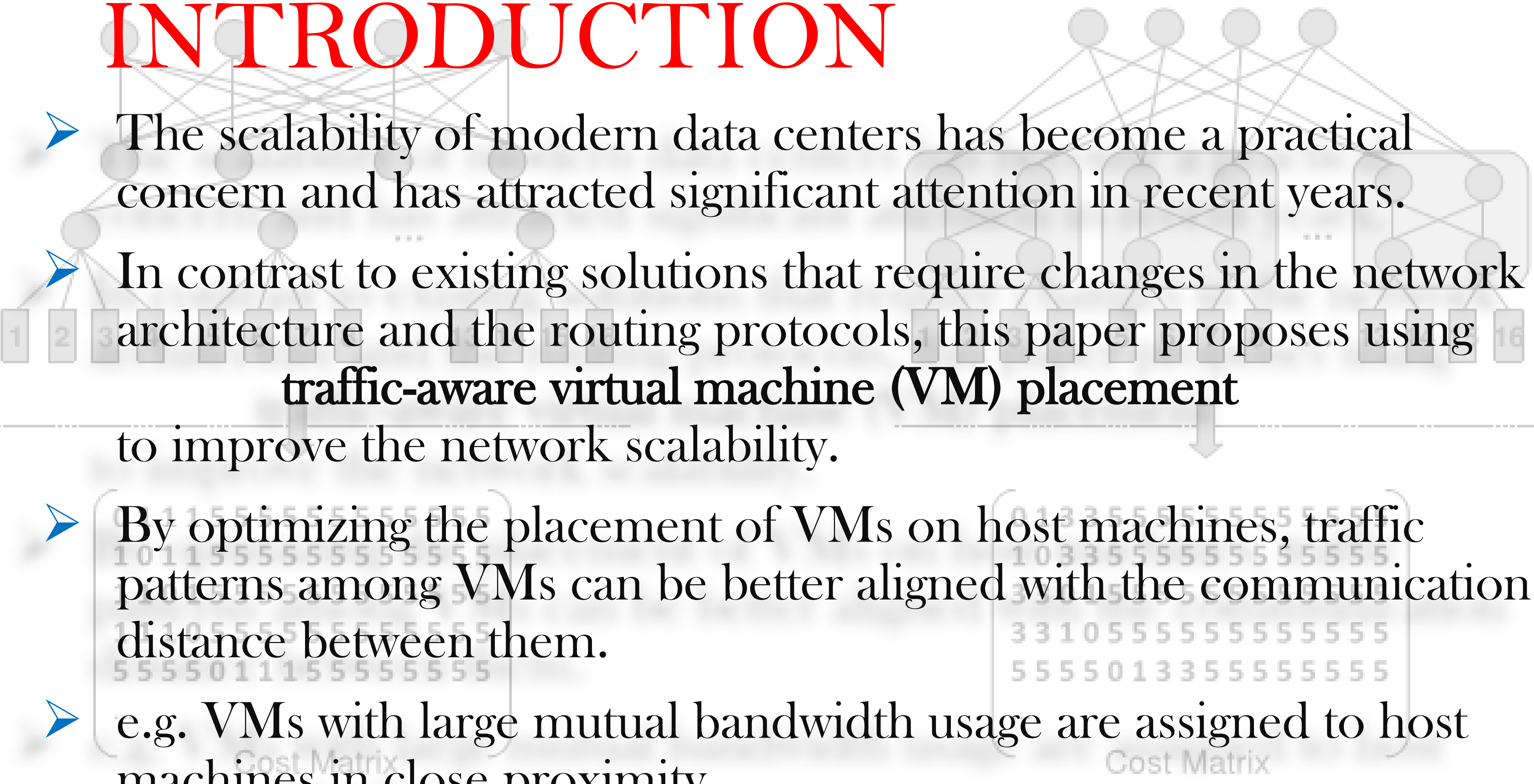


Overview:

- INTRODUCTION
- BACKGROUND
- VIRTUAL MACHINE PLACEMENT PROBLEM
- ALGORITHMS
- IMPACT OF NETWORK ARCHITECTURES AND TRAFFIC PATTERNS ON OPTIMAL VM PLACEMENTS
- EVALUATION OF ALGORITHM CLUSTER-AND-CUT
- DISCUSSION AND FUTURE WORK



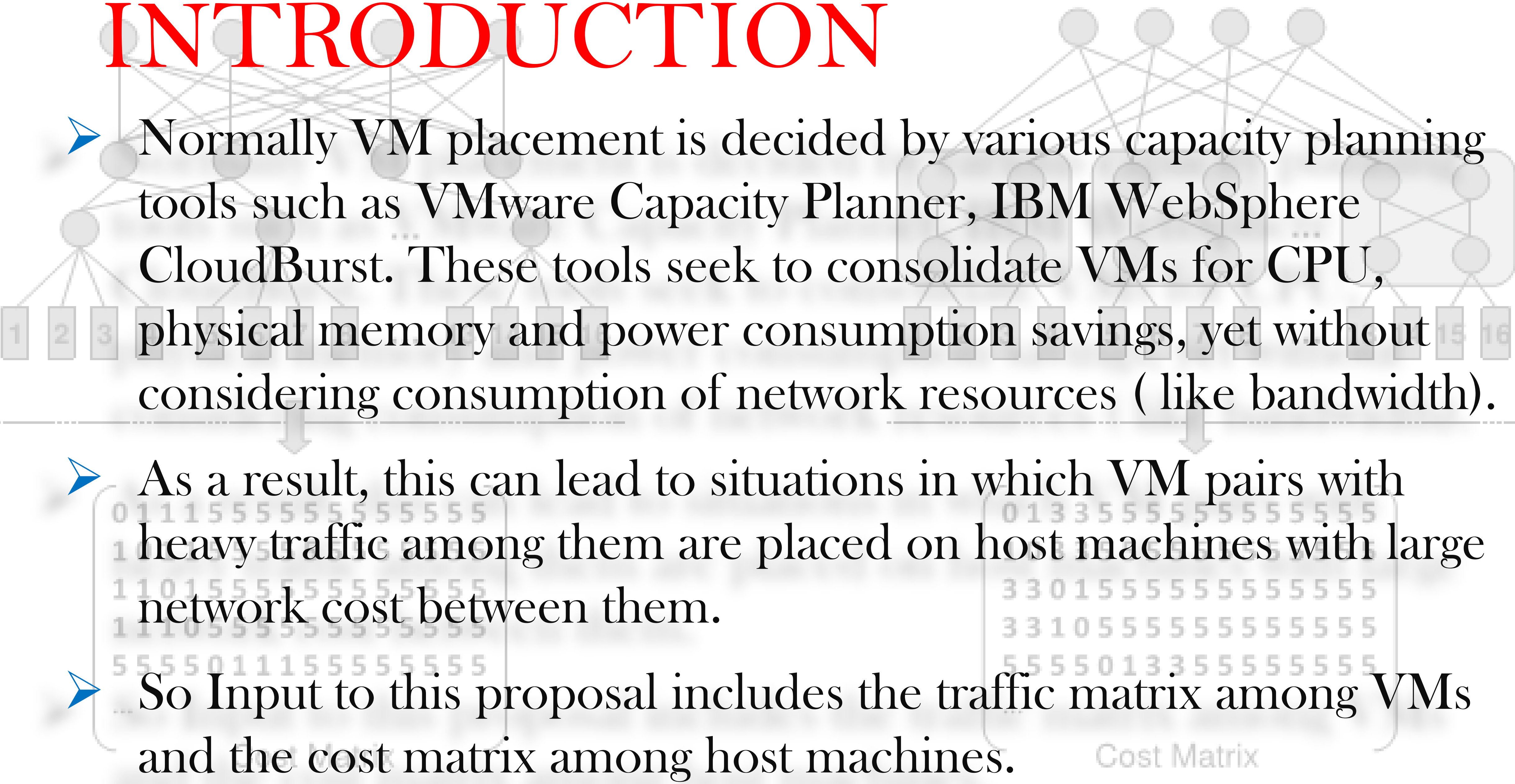
INTRODUCTION



- The scalability of modern data centers has become a practical concern and has attracted significant attention in recent years.
- In contrast to existing solutions that require changes in the network architecture and the routing protocols, this paper proposes using **traffic-aware virtual machine (VM) placement** to improve the network scalability.
- By optimizing the placement of VMs on host machines, traffic patterns among VMs can be better aligned with the communication distance between them.
- e.g. VMs with large mutual bandwidth usage are assigned to host machines in close proximity

INTRODUCTION

- Normally VM placement is decided by various capacity planning tools such as VMware Capacity Planner, IBM WebSphere CloudBurst. These tools seek to consolidate VMs for CPU, physical memory and power consumption savings, yet without considering consumption of network resources (like bandwidth).
- As a result, this can lead to situations in which VM pairs with heavy traffic among them are placed on host machines with large network cost between them.
- So Input to this proposal includes the traffic matrix among VMs and the cost matrix among host machines.

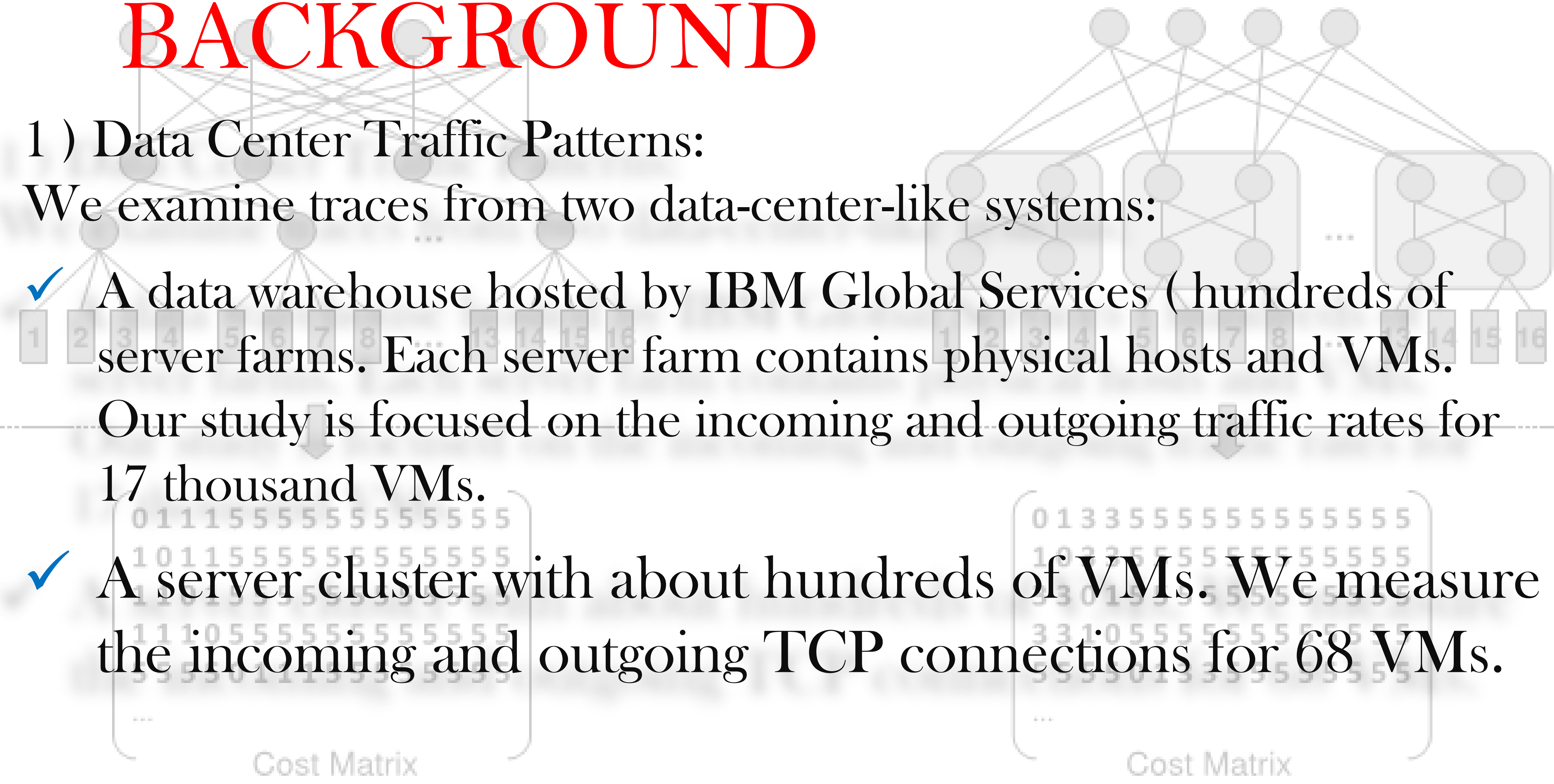


BACKGROUND

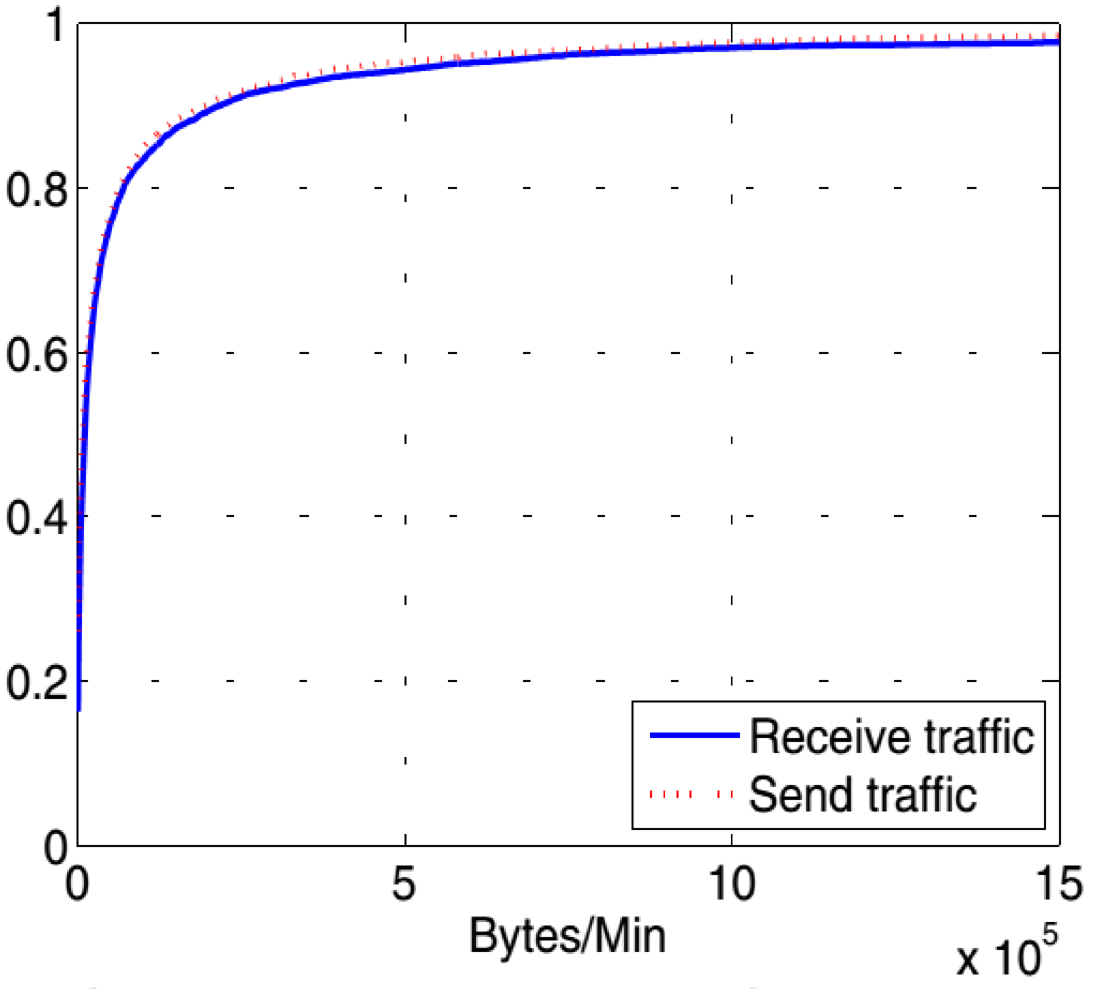
1) Data Center Traffic Patterns:

We examine traces from two data-center-like systems:

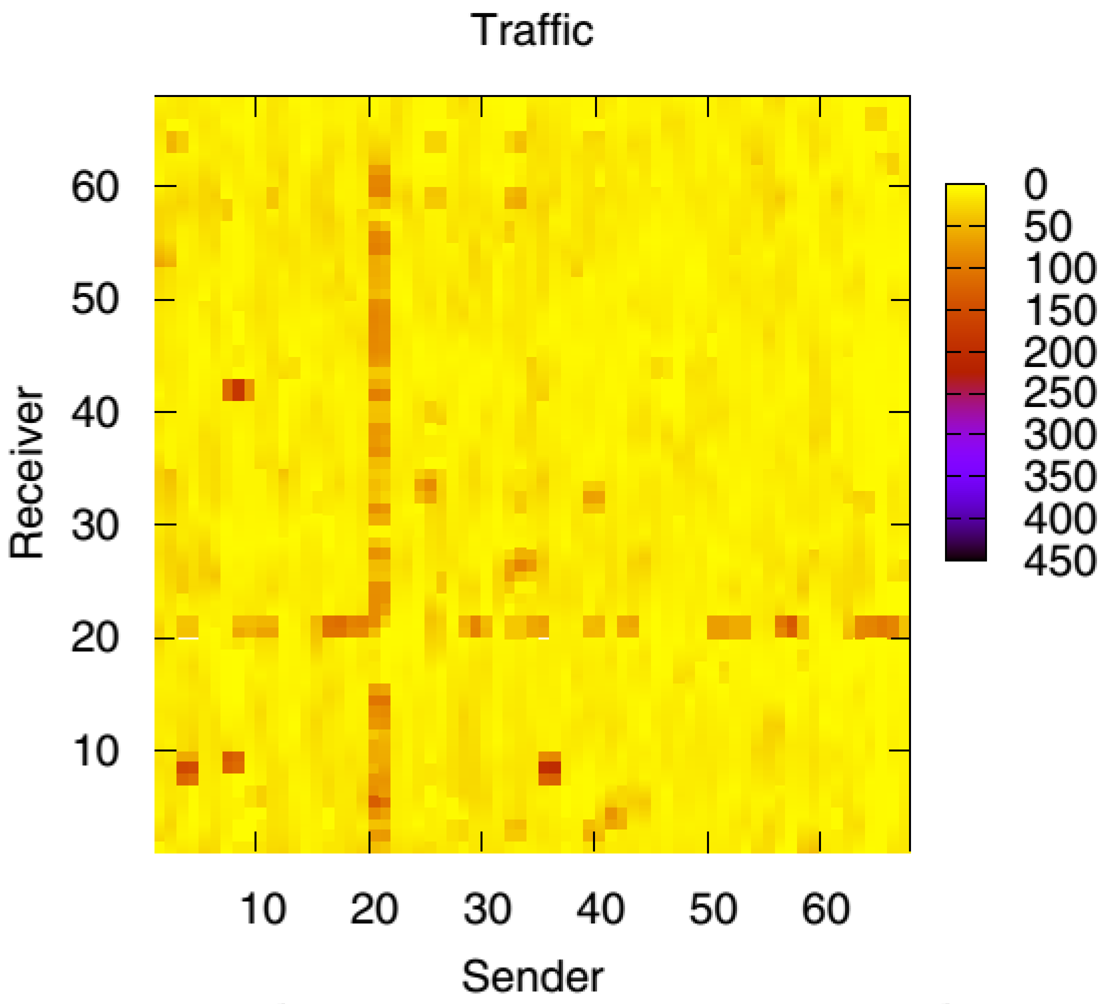
- ✓ A data warehouse hosted by IBM Global Services (hundreds of server farms. Each server farm contains physical hosts and VMs. Our study is focused on the incoming and outgoing traffic rates for 17 thousand VMs.
- ✓ A server cluster with about hundreds of VMs. We measure the incoming and outgoing TCP connections for 68 VMs.



BACKGROUND



Cost Matrix



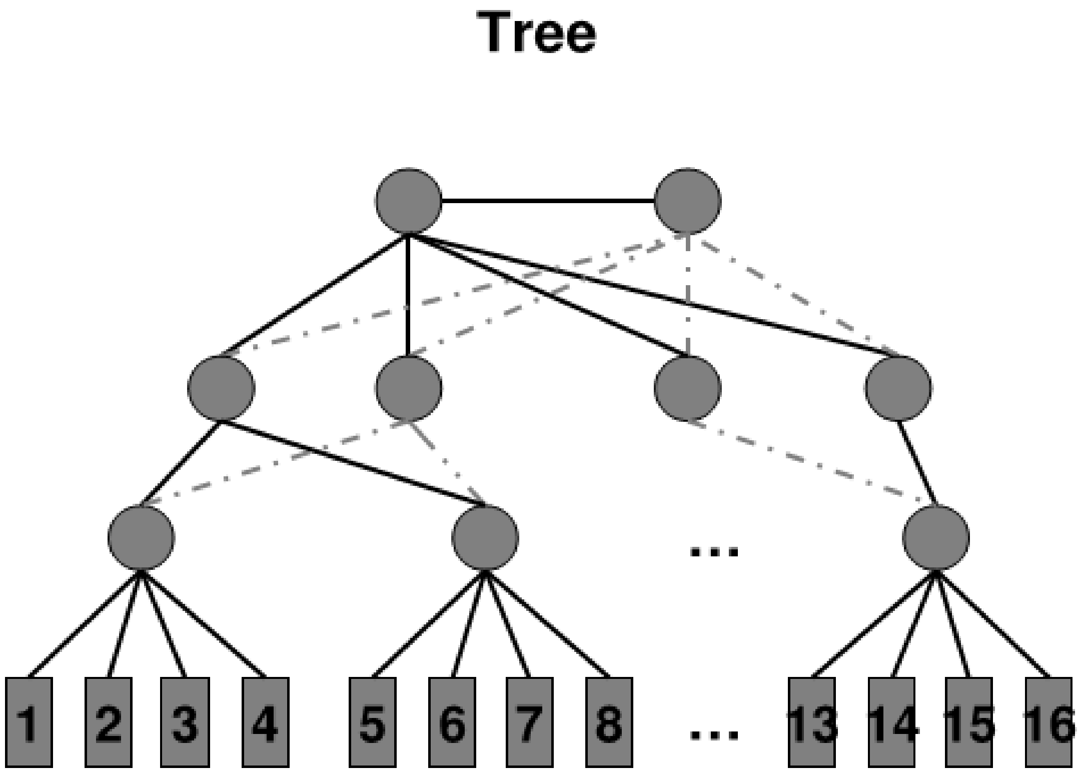
Cost Matrix

BACKGROUND

2) Data Center Network Architectures:

Three-tier architecture: the access tier, aggregation tier, core tier.

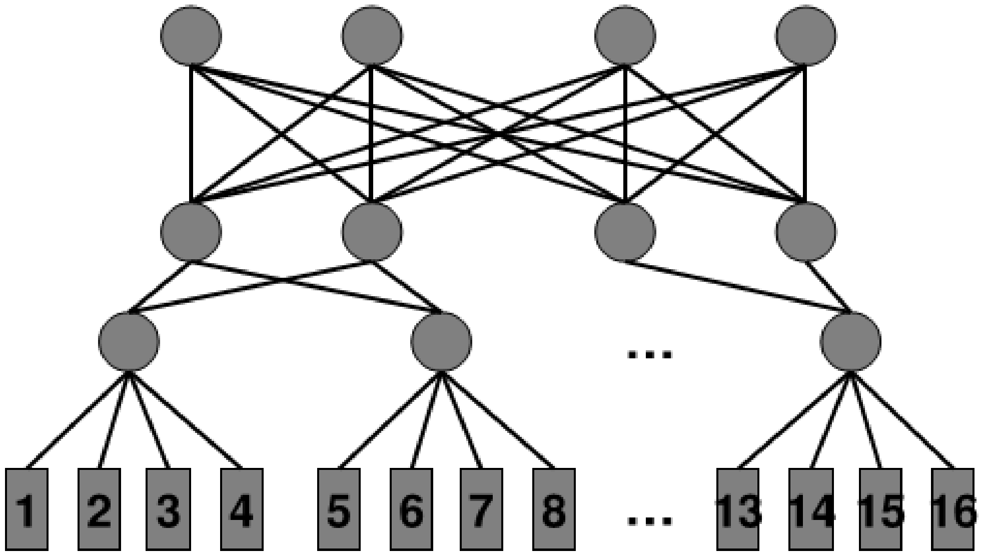
- ✓ Tree:



BACKGROUND

- ✓ VL2: Shares many features with the Tree, but:
  - The core tier and the aggregation tier form a Clos topology, i.e. the aggregation switches are connected with the core ones by forming a complete bipartite graph.
  - Traffic originated from the access switches is forwarded in the aggregation and the core tiers, i.e. it is forwarded first to a randomly selected core switch and then back to the actual destination.

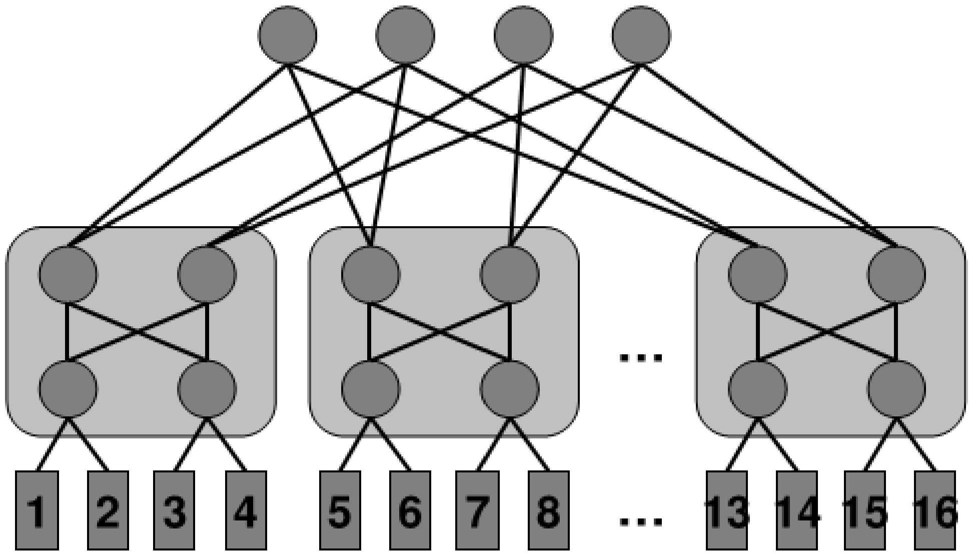
VL2



BACKGROUND

- ✓ Fat-Tree(PortLand): It is built around the concept of pods: a collection of access and aggregation switches that forma complete bipartite graph, i.e., a Clos graph.
  - Each pod is connected with all core switches, by evenly distributing the up-links between all the aggregation switches of the pod. As such, a second Clos topology is generated between the core switches and the pods.
  - PortLand assumes all switches are identical, i.e., they have the same number of ports (something not required by the previous ones)

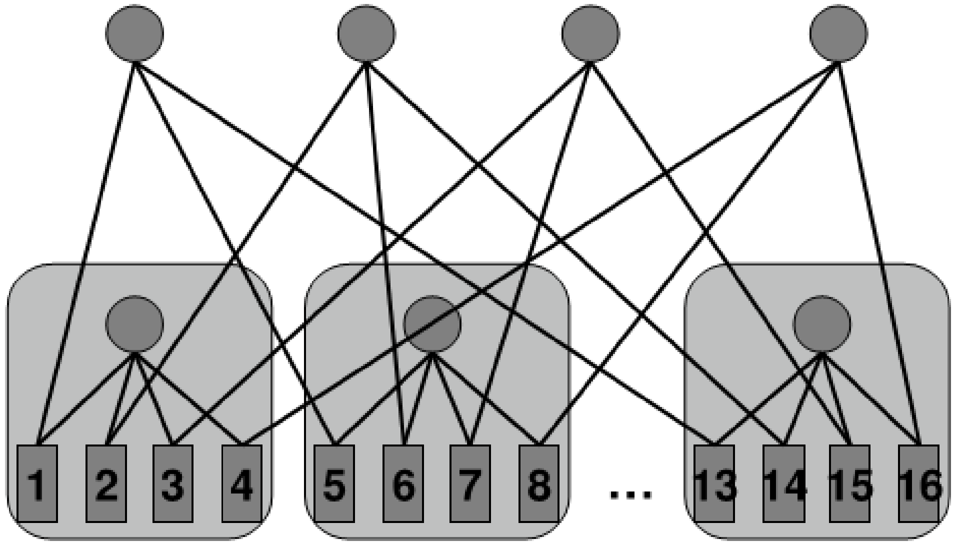
Fat-Tree



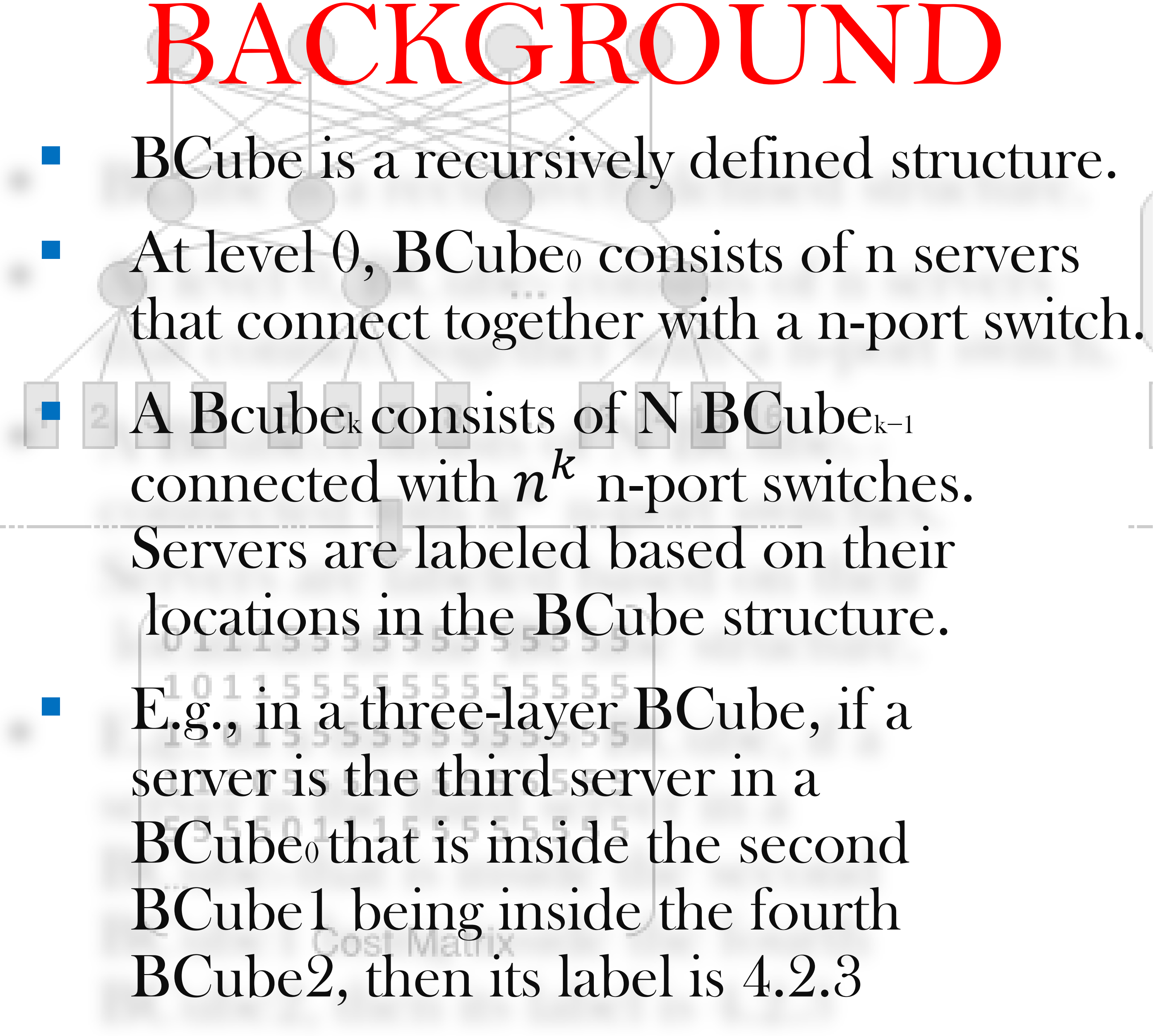
BACKGROUND

- ✓ BCube: a new multi-level network architecture for the data center with the following distinguishing feature:
  - Servers are part of the network infrastructure, i.e., they forward packets on behalf of other servers.

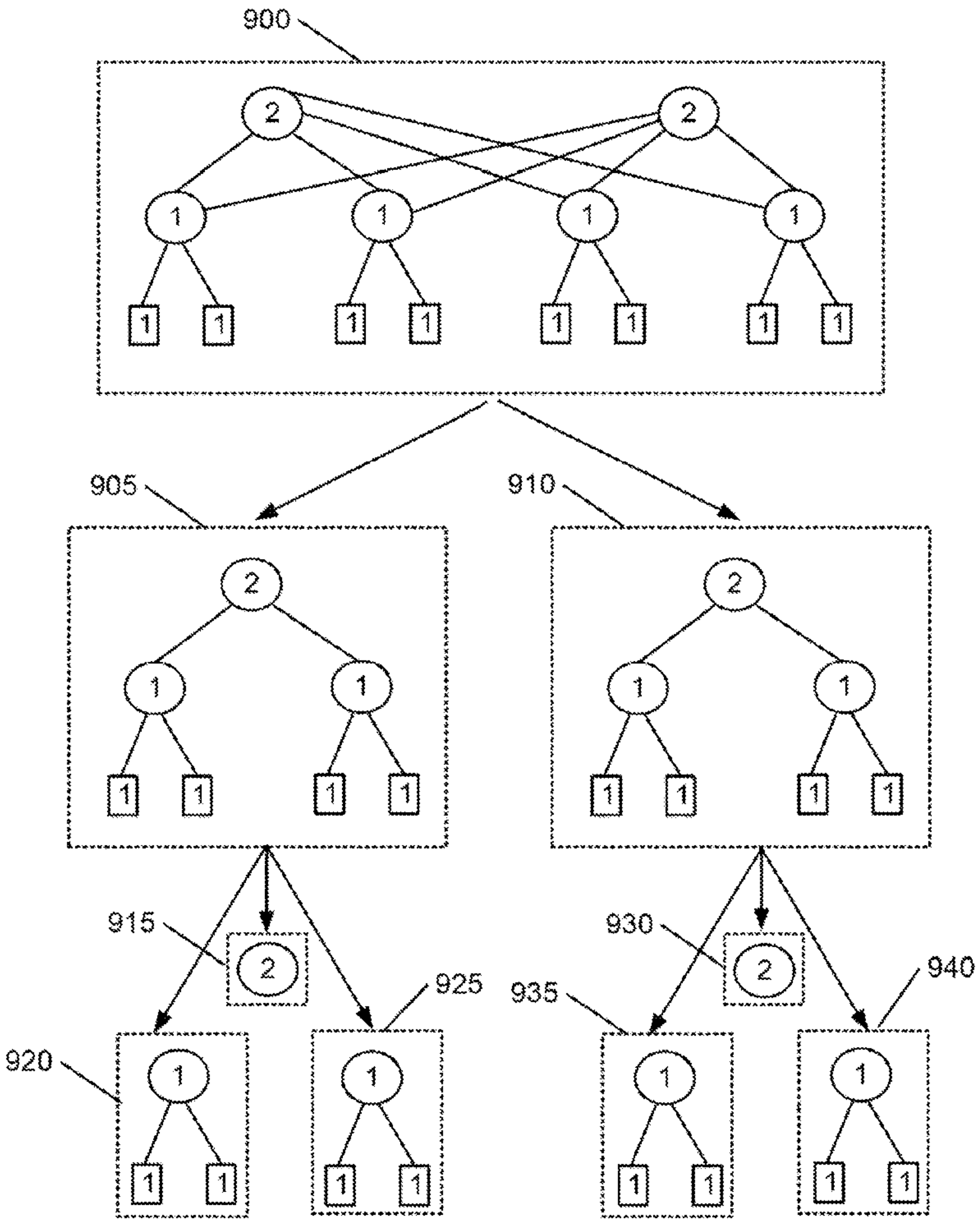
BCube



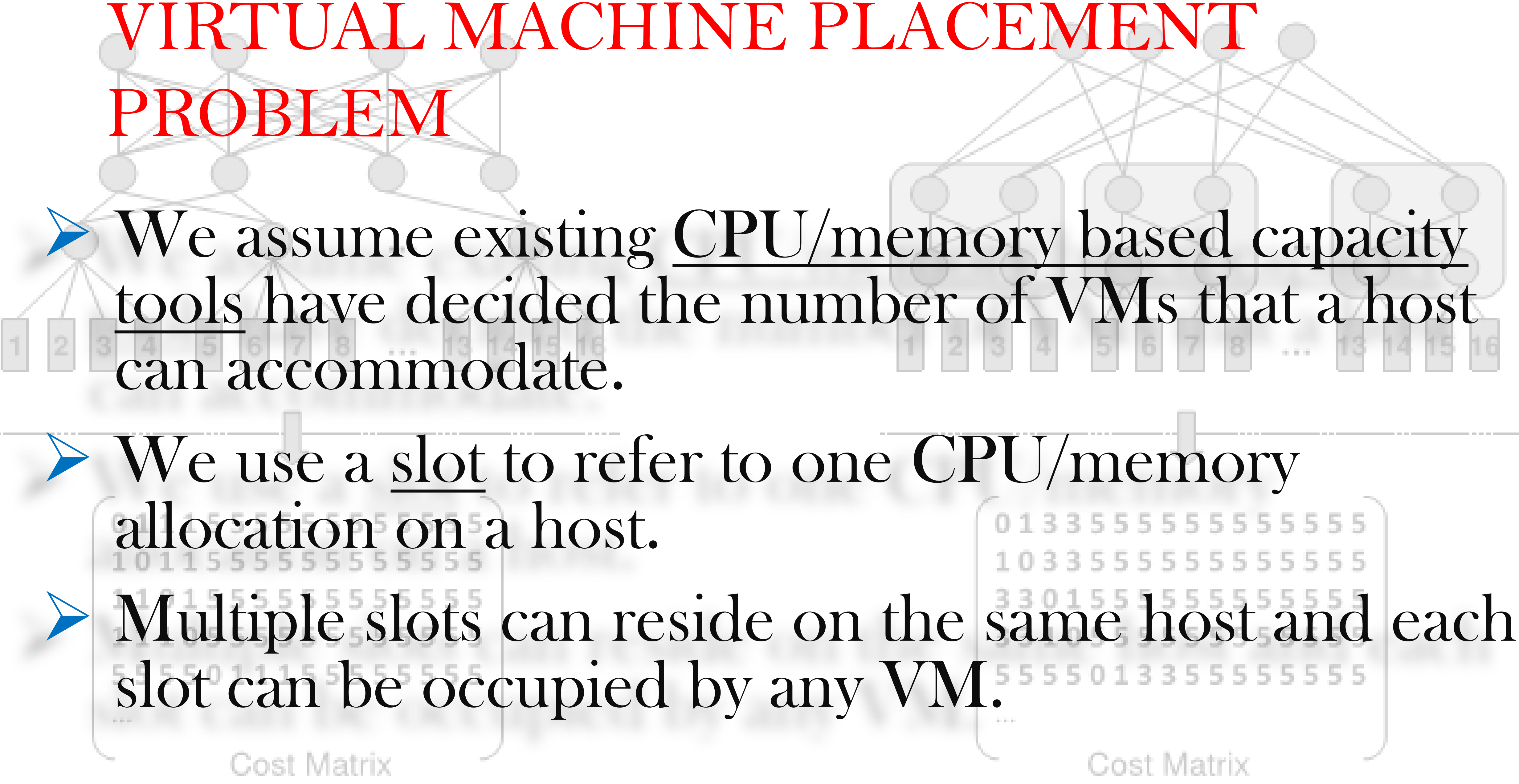
BACKGROUND



- BCube is a recursively defined structure.
- At level 0, BCube$_0$ consists of n servers that connect together with a n-port switch.
- A Bcube$_k$ consists of N BCube$_{k-1}$ connected with n^k n-port switches. Servers are labeled based on their locations in the BCube structure.
- E.g., in a three-layer BCube, if a server is the third server in a BCube$_0$ that is inside the second BCube1 being inside the fourth BCube2, then its label is 4.2.3

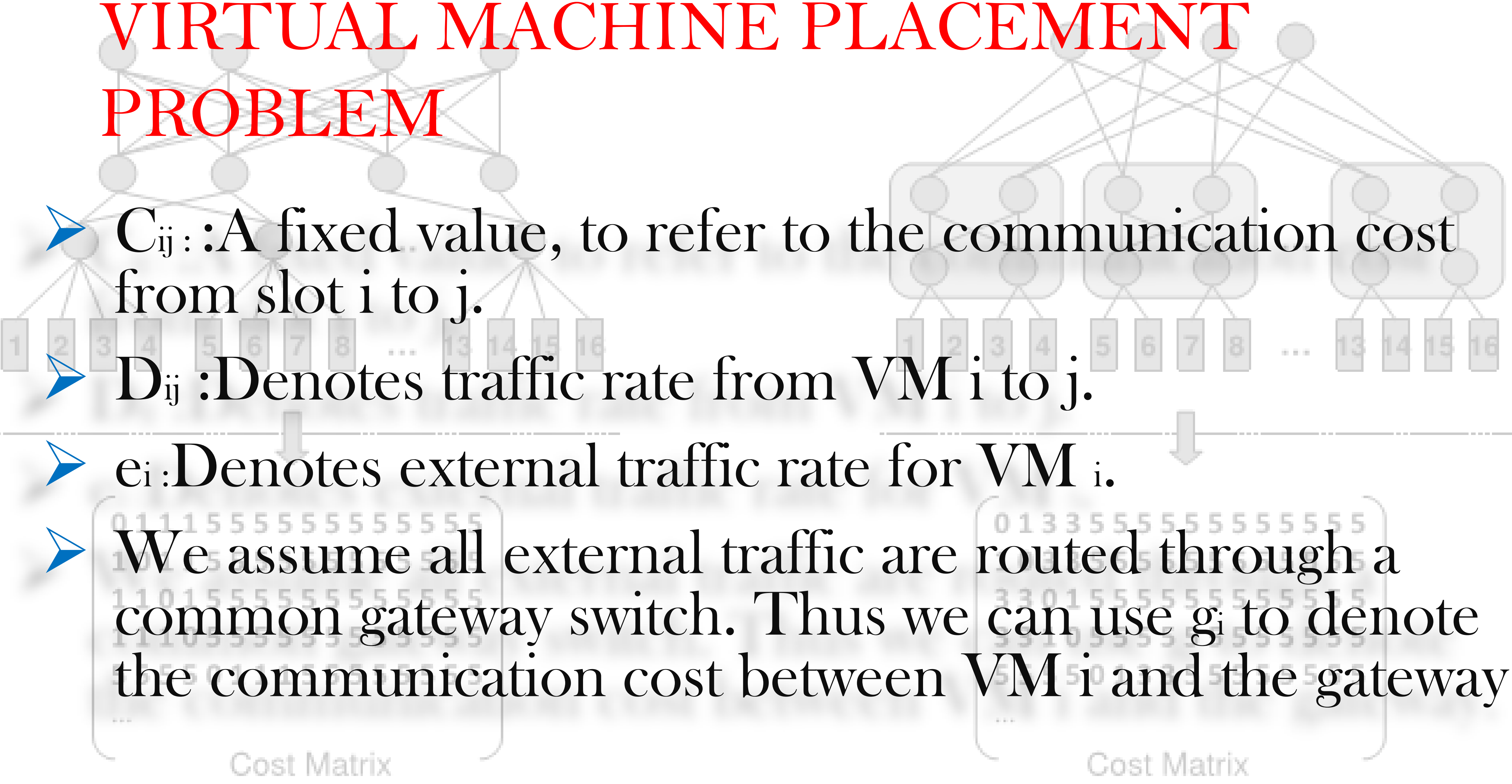


VIRTUAL MACHINE PLACEMENT PROBLEM



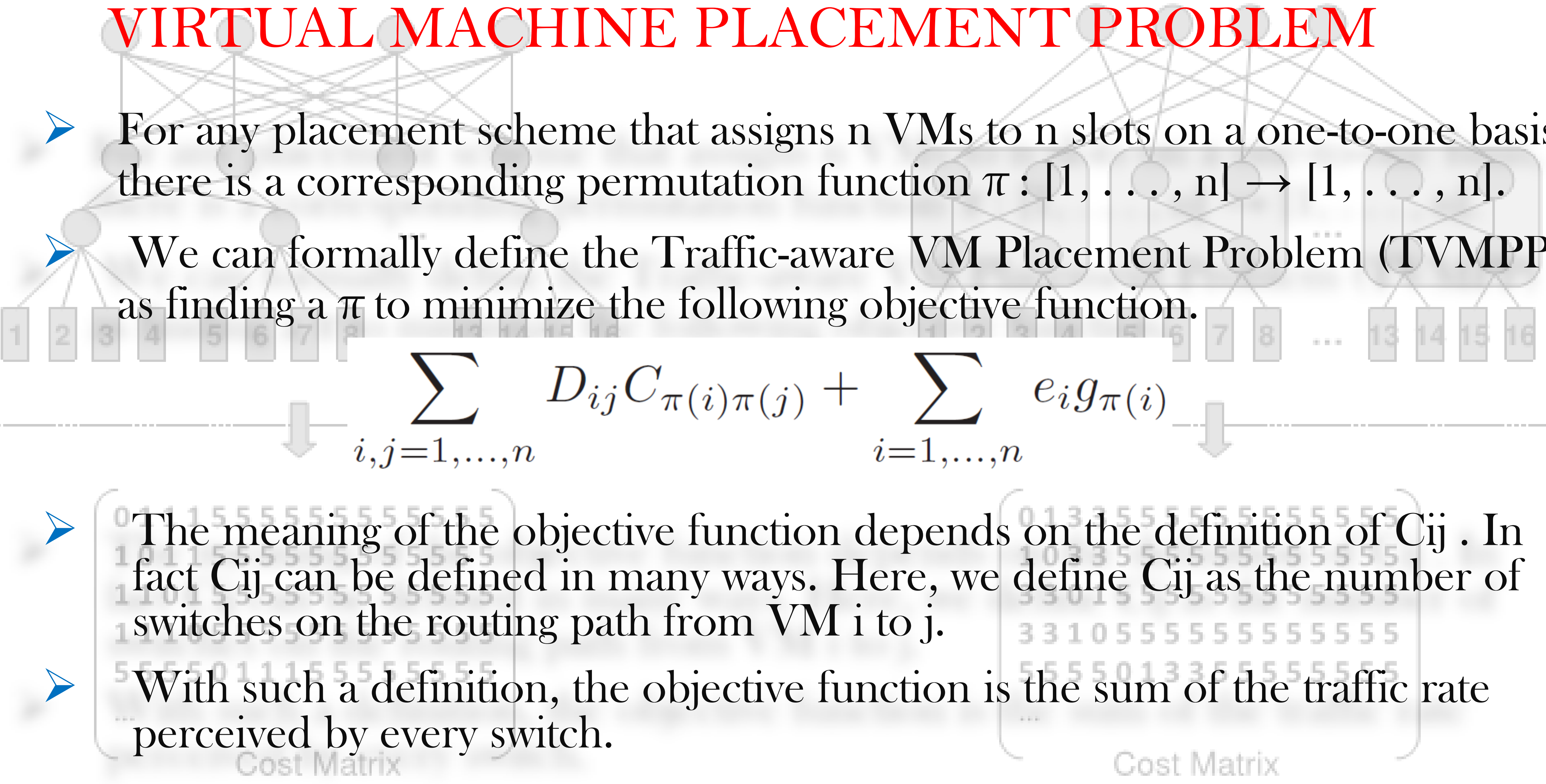
- We assume existing CPU/memory based capacity tools have decided the number of VMs that a host can accommodate.
- We use a slot to refer to one CPU/memory allocation on a host.
- Multiple slots can reside on the same host and each slot can be occupied by any VM.

VIRTUAL MACHINE PLACEMENT PROBLEM



- C_{ij} : :A fixed value, to refer to the communication cost from slot i to j.
- D_{ij} :Denotes traffic rate from VM i to j.
- e_i :Denotes external traffic rate for VM $_i$.
- We assume all external traffic are routed through a common gateway switch. Thus we can use g_i to denote the communication cost between VM i and the gateway.

VIRTUAL MACHINE PLACEMENT PROBLEM



- For any placement scheme that assigns n VMs to n slots on a one-to-one basis, there is a corresponding permutation function π : [1, . . . , n] → [1, . . . , n].
- We can formally define the Traffic-aware VM Placement Problem (TVMPP) as finding a π to minimize the following objective function.

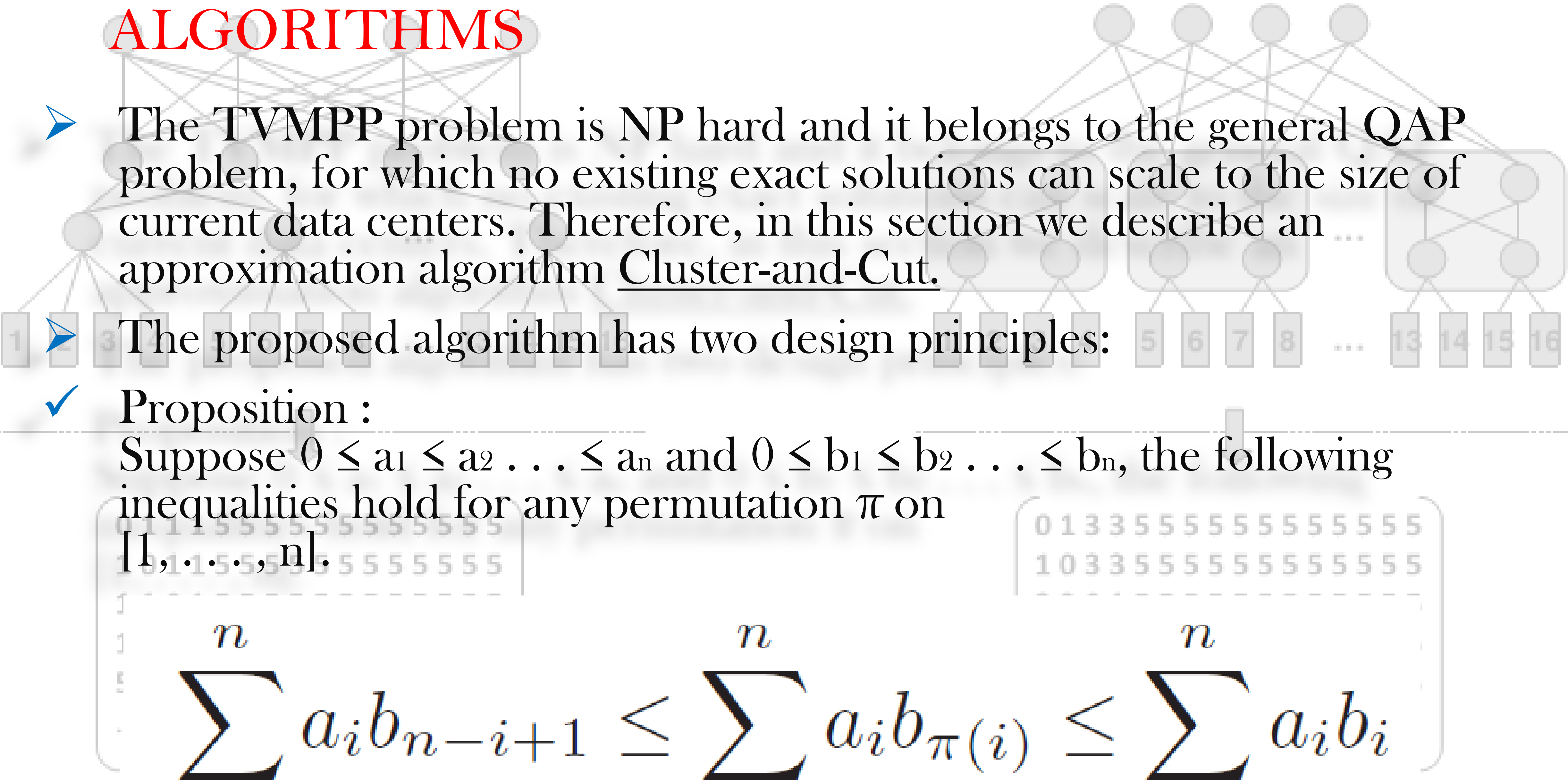
$$\sum_{i,j=1,\ldots,n} D_{ij} C_{\pi(i)\pi(j)} + \sum_{i=1,\ldots,n} e_i g_{\pi(i)}$$

- The meaning of the objective function depends on the definition of Cij . In fact Cij can be defined in many ways. Here, we define Cij as the number of switches on the routing path from VM i to j.
- With such a definition, the objective function is the sum of the traffic rate perceived by every switch.

VIRTUAL MACHINE PLACEMENT PROBLEM

- If the objective function is normalized by the sum of VM-to-VM bandwidth demand, it is equivalent to the average number of switches that a data unit traverses.
- If we further assume every switch causes equal delay, the objective function can be interpreted as the average latency for a data unit traversing the network.
- Accordingly, optimizing TVMPP is equivalent to minimizing average traffic latency caused by network infrastructure.
- Notice that the second part in the objective function is the total external traffic rate calculated at all switches. In reality, this sum is most likely constant regardless of VM placement, because in typical data center networks, the cost between every end host and the gateway is the same. Therefore, the second part in the objective function can be ignored in our analysis.
- When C and D are matrices with arbitrary real values, TVMPP falls into the category of Quadratic Assignment Problem (QAP). QAP is a known NP-hard problem.

ALGORITHMS

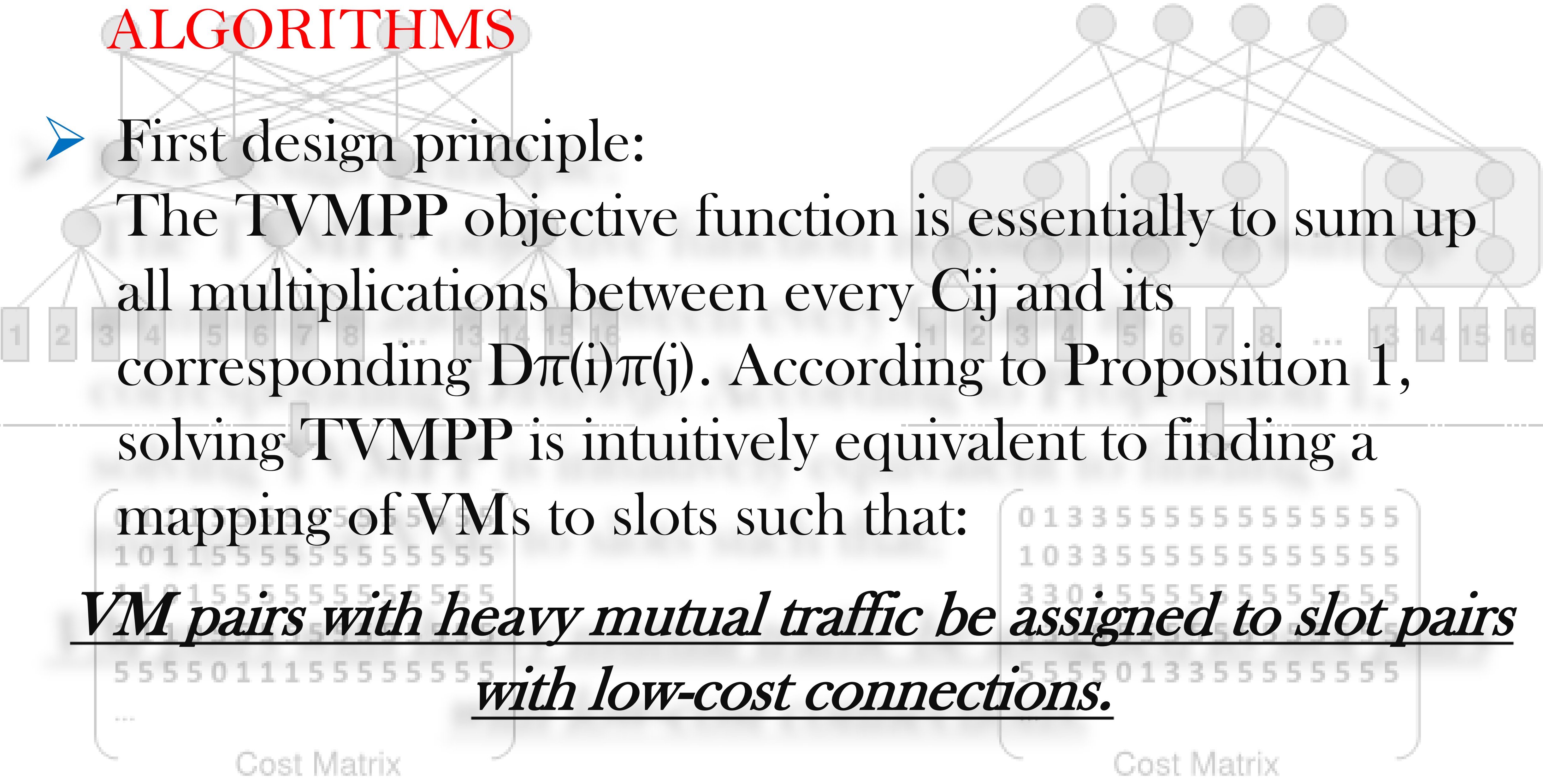


- The TVMPP problem is NP hard and it belongs to the general QAP problem, for which no existing exact solutions can scale to the size of current data centers. Therefore, in this section we describe an approximation algorithm Cluster-and-Cut.
- The proposed algorithm has two design principles:

✓ Proposition :
Suppose $0 \le a_1 \le a_2 \ldots \le a_n$ and $0 \le b_1 \le b_2 \ldots \le b_n$, the following inequalities hold for any permutation π on $[1, \ldots, n]$.

$$\sum_{i=1}^{n} a_i b_{n-i+1} \le \sum_{i=1}^{n} a_i b_{\pi(i)} \le \sum_{i=1}^{n} a_i b_i$$

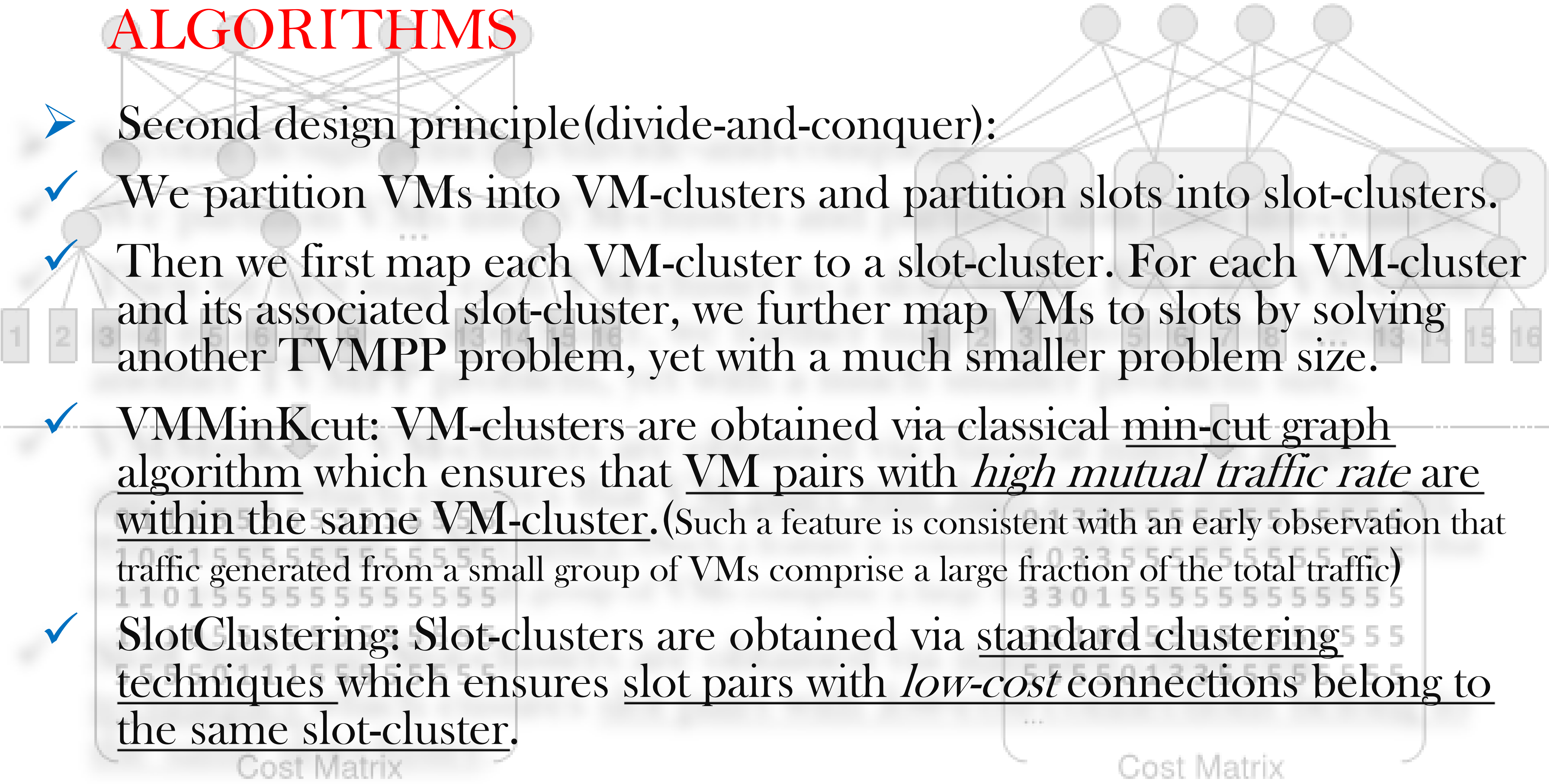
ALGORITHMS



- First design principle:
  The TVMPP objective function is essentially to sum up all multiplications between every Cij and its corresponding Dπ(i)π(j). According to Proposition 1, solving TVMPP is intuitively equivalent to finding a mapping of VMs to slots such that:

VM pairs with heavy mutual traffic be assigned to slot pairs with low-cost connections.

ALGORITHMS



➢ Second design principle(divide-and-conquer):

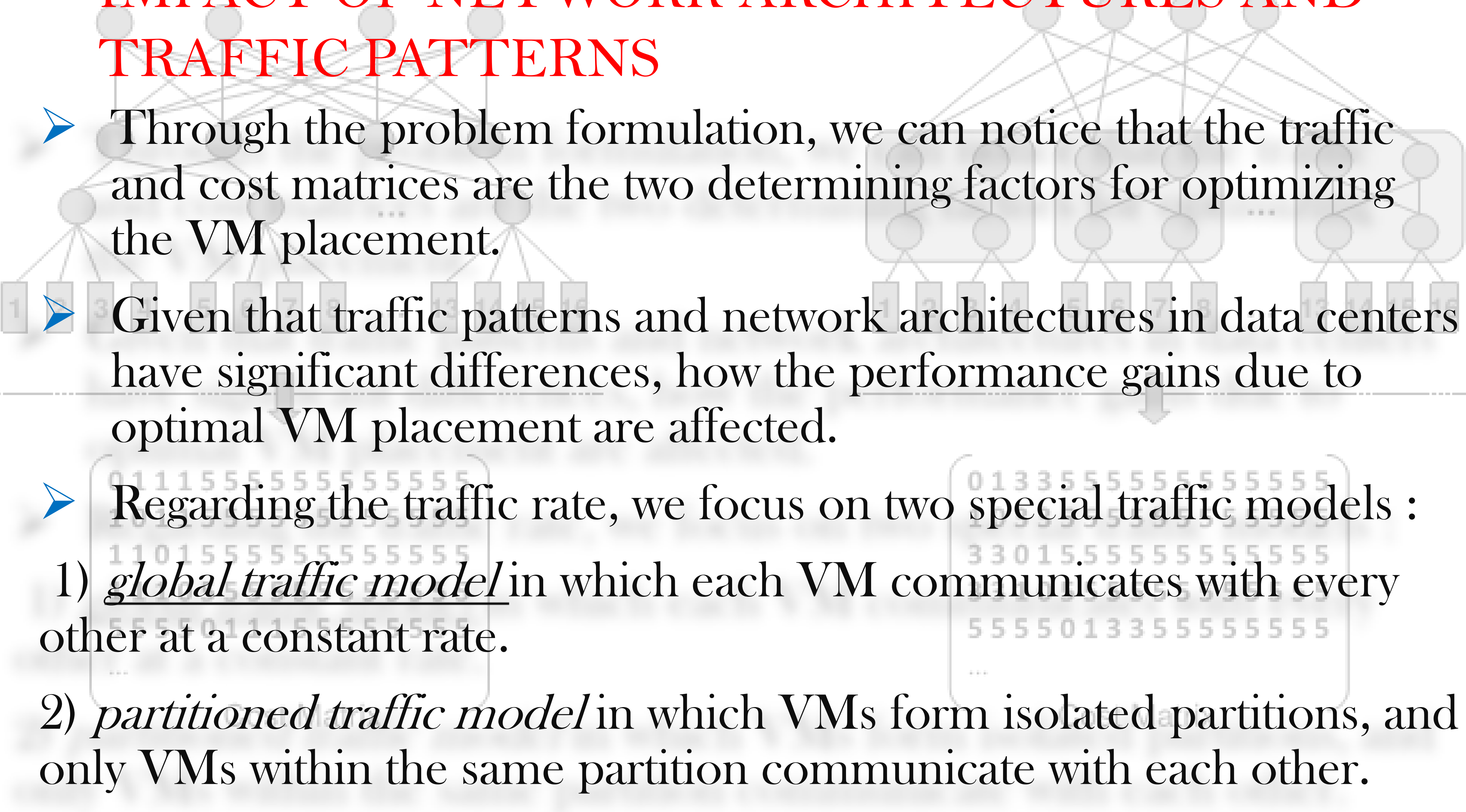
✓ We partition VMs into VM-clusters and partition slots into slot-clusters.

✓ Then we first map each VM-cluster to a slot-cluster. For each VM-cluster and its associated slot-cluster, we further map VMs to slots by solving another TVMPP problem, yet with a much smaller problem size.

✓ VMMinKcut: VM-clusters are obtained via classical min-cut graph algorithm which ensures that VM pairs with *high mutual traffic rate* are within the same VM-cluster.(Such a feature is consistent with an early observation that traffic generated from a small group of VMs comprise a large fraction of the total traffic)

✓ SlotClustering: Slot-clusters are obtained via standard clustering techniques which ensures slot pairs with *low-cost* connections belong to the same slot-cluster.

IMPACT OF NETWORK ARCHITECTURES AND TRAFFIC PATTERNS



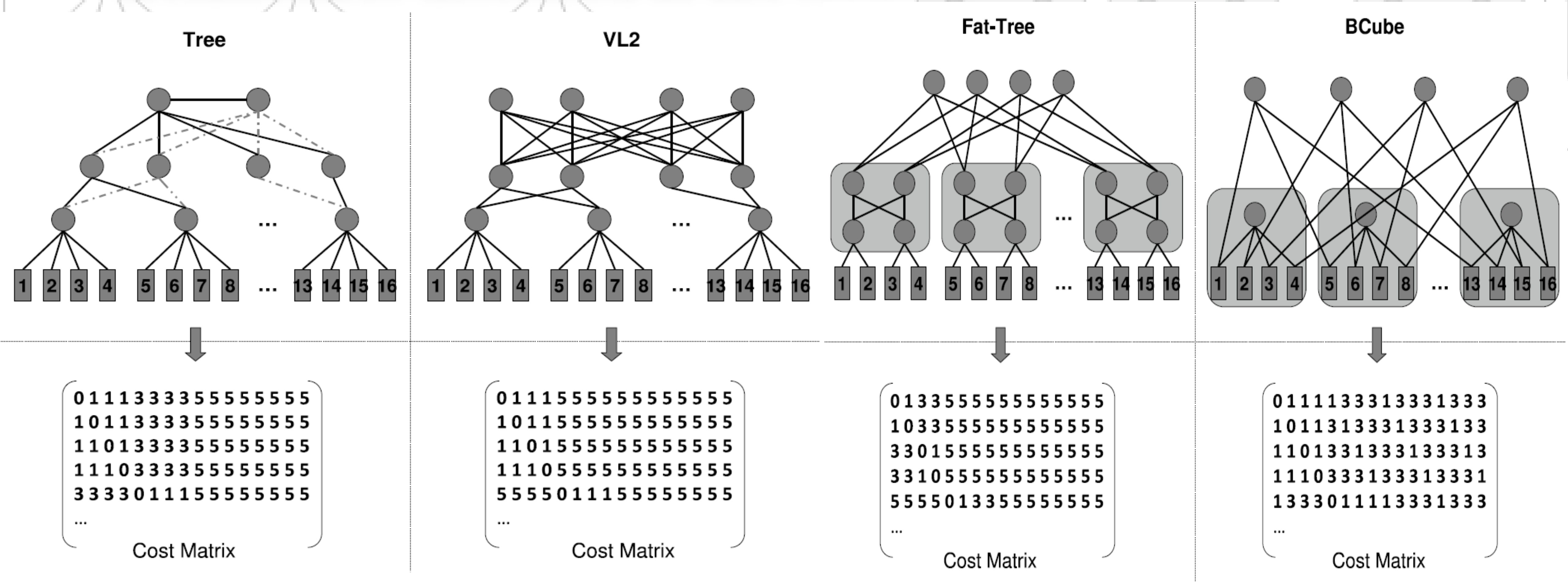
- Through the problem formulation, we can notice that the traffic and cost matrices are the two determining factors for optimizing the VM placement.
- Given that traffic patterns and network architectures in data centers have significant differences, how the performance gains due to optimal VM placement are affected.
- Regarding the traffic rate, we focus on two special traffic models :

1) *global traffic model* in which each VM communicates with every other at a constant rate.

2) *partitioned traffic model* in which VMs form isolated partitions, and only VMs within the same partition communicate with each other.

IMPACT OF NETWORK ARCHITECTURES AND TRAFFIC PATTERNS

- Regarding network architectures (cost), we focus on the four architectures described in last section.



Tree

$$\begin{pmatrix}
0&1&1&1&3&3&3&3&5&5&5&5&5&5&5&5\\
1&0&1&1&3&3&3&3&5&5&5&5&5&5&5&5\\
1&1&0&1&3&3&3&3&5&5&5&5&5&5&5&5\\
1&1&1&0&3&3&3&3&5&5&5&5&5&5&5&5\\
3&3&3&3&0&1&1&1&5&5&5&5&5&5&5&5\\
\dots
\end{pmatrix}$$

Cost Matrix

VL2

$$\begin{pmatrix}
0&1&1&1&5&5&5&5&5&5&5&5&5&5&5&5\\
1&0&1&1&5&5&5&5&5&5&5&5&5&5&5&5\\
1&1&0&1&5&5&5&5&5&5&5&5&5&5&5&5\\
1&1&1&0&5&5&5&5&5&5&5&5&5&5&5&5\\
5&5&5&5&0&1&1&1&5&5&5&5&5&5&5&5\\
\dots
\end{pmatrix}$$

Cost Matrix

Fat-Tree

$$\begin{pmatrix}
0&1&3&3&5&5&5&5&5&5&5&5&5&5&5&5\\
1&0&3&3&5&5&5&5&5&5&5&5&5&5&5&5\\
3&3&0&1&5&5&5&5&5&5&5&5&5&5&5&5\\
3&3&1&0&5&5&5&5&5&5&5&5&5&5&5&5\\
5&5&5&5&0&1&3&3&5&5&5&5&5&5&5&5\\
\dots
\end{pmatrix}$$

Cost Matrix

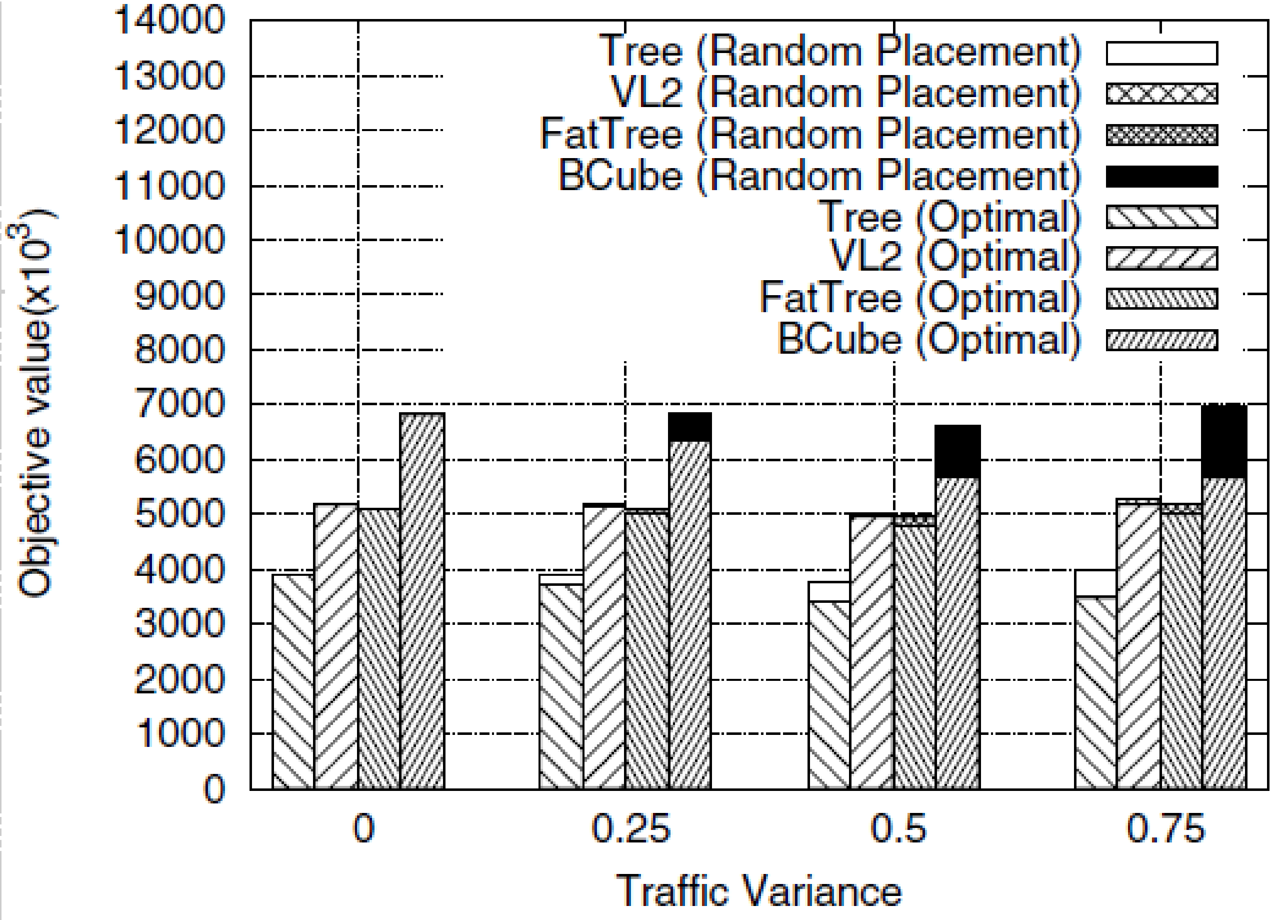
BCube

$$\begin{pmatrix}
0&1&1&1&1&3&3&3&1&3&3&3&1&3&3&3\\
1&0&1&1&3&1&3&3&3&1&3&3&3&1&3&3\\
1&1&0&1&3&3&1&3&3&3&1&3&3&3&1&3\\
1&1&1&0&3&3&3&1&3&3&3&1&3&3&3&1\\
1&3&3&3&0&1&1&1&1&3&3&3&1&3&3&3\\
\dots
\end{pmatrix}$$

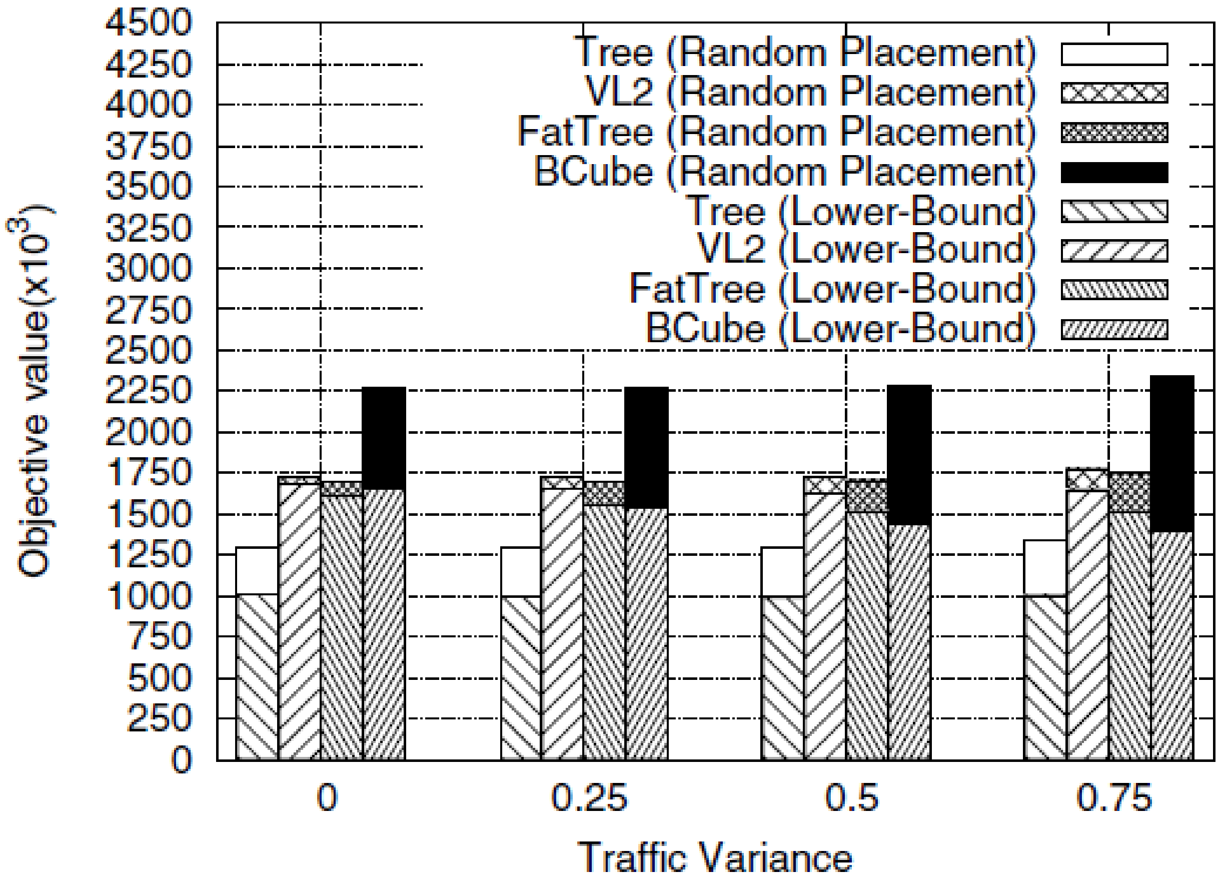
Cost Matrix

IMPACT OF NETWORK ARCHITECTURES AND TRAFFIC PATTERNS

Global traffic model

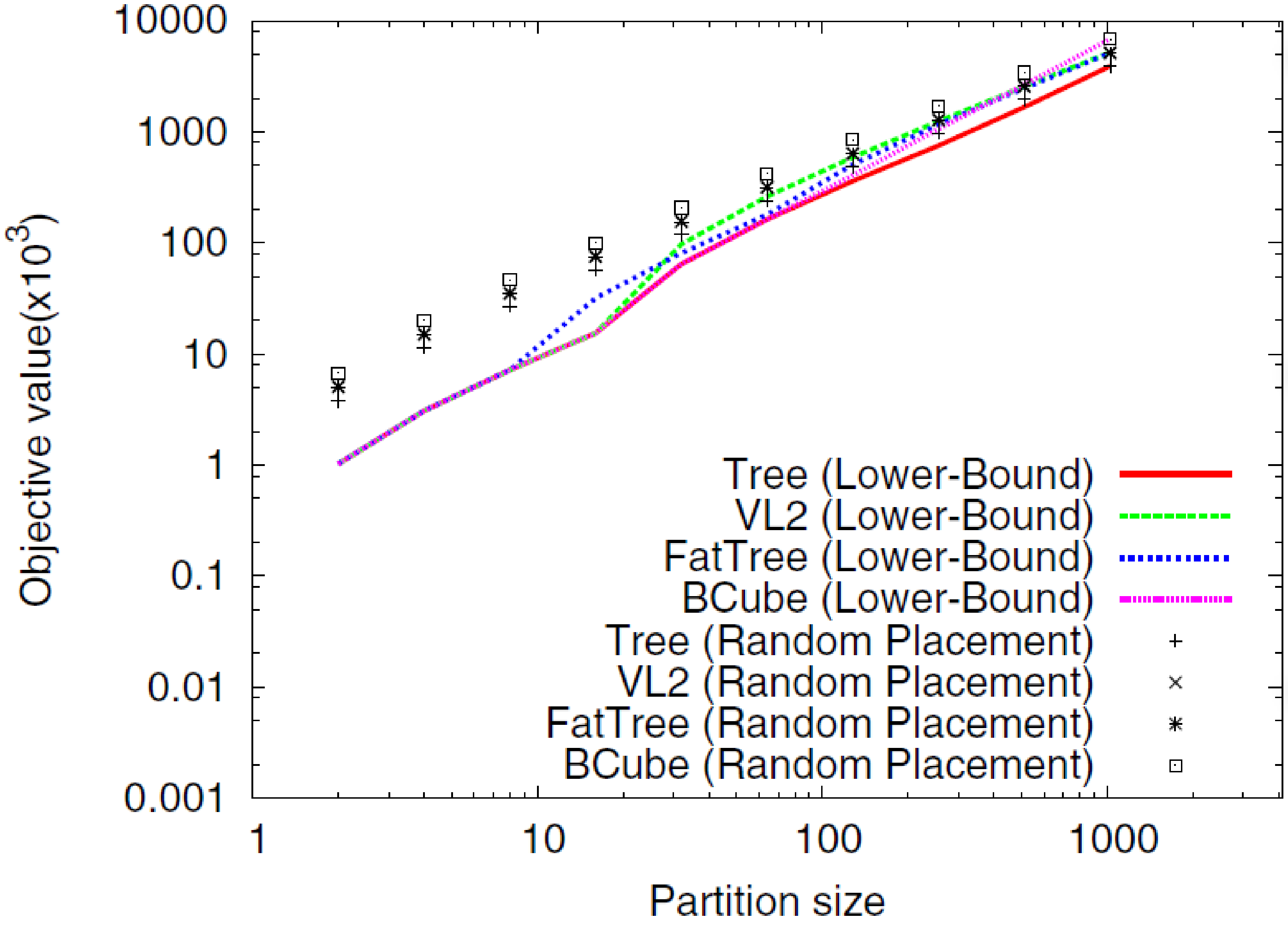


Partitioned traffic model



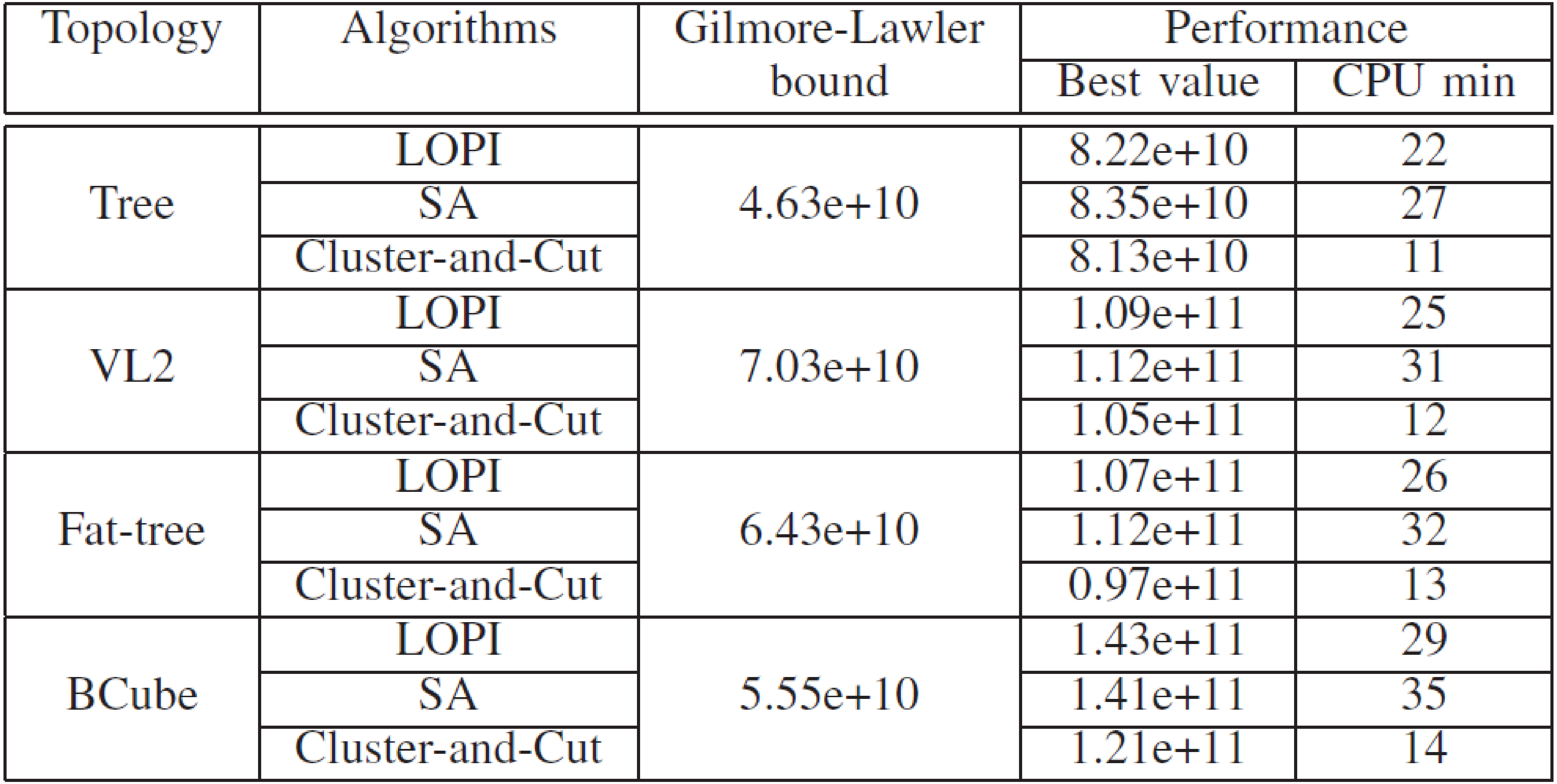
IMPACT OF NETWORK ARCHITECTURES AND TRAFFIC PATTERNS

Different partition size



- **Summary:**
  - ✓ The potential benefit of optimizing TVMPP is greater with increased traffic variance within one partition.
  - ✓ The potential benefit of optimizing TVMPP is greater with increased number of traffic partitions.
  - ✓ The potential benefit of optimizing TVMPP depends on the network architecture.

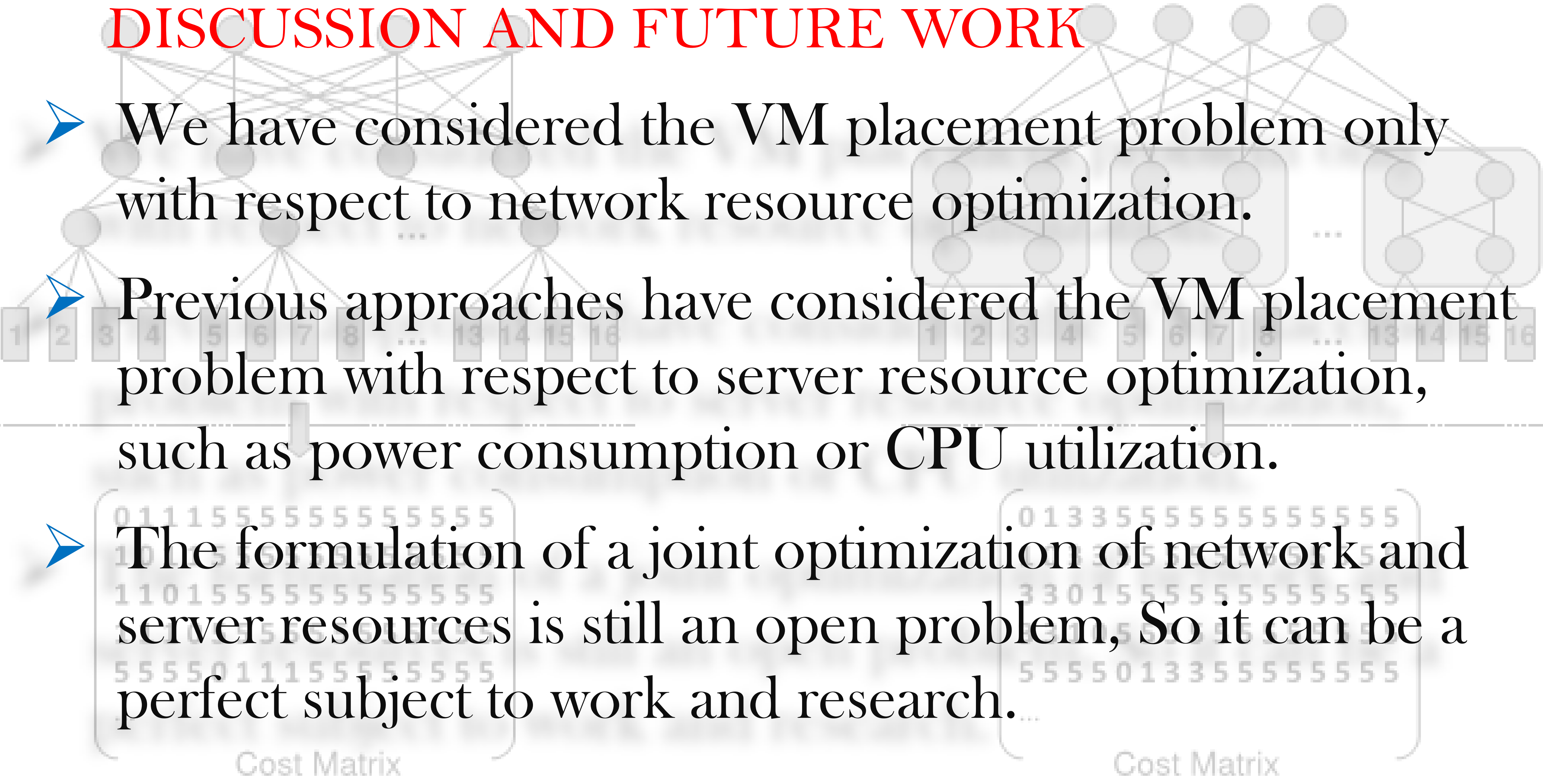
EVALUATION OF ALGORITHM CLUSTER-AND-CUT



Topology	Algorithms	Gilmore-Lawler bound	Performance	
			Best value	CPU min
Tree	LOPI	4.63e+10	8.22e+10	22
	SA		8.35e+10	27
	Cluster-and-Cut		8.13e+10	11
VL2	LOPI	7.03e+10	1.09e+11	25
	SA		1.12e+11	31
	Cluster-and-Cut		1.05e+11	12
Fat-tree	LOPI	6.43e+10	1.07e+11	26
	SA		1.12e+11	32
	Cluster-and-Cut		0.97e+11	13
BCube	LOPI	5.55e+10	1.43e+11	29
	SA		1.41e+11	35
	Cluster-and-Cut		1.21e+11	14

DISCUSSION AND FUTURE WORK

- We have considered the VM placement problem only with respect to network resource optimization.
- Previous approaches have considered the VM placement problem with respect to server resource optimization, such as power consumption or CPU utilization.
- The formulation of a joint optimization of network and server resources is still an open problem, So it can be a perfect subject to work and research.



THANKS FOR YOUR ATTENTION