

An Optimization Problem in Adaptive Virtual Environments

Ananth I. Sundararaj

Manan Sanghi

Jack R. Lange

Peter A. Dinda



Prescience Lab

Department of Computer Science

Northwestern University

<http://virtuoso.cs.northwestern.edu>

Summary

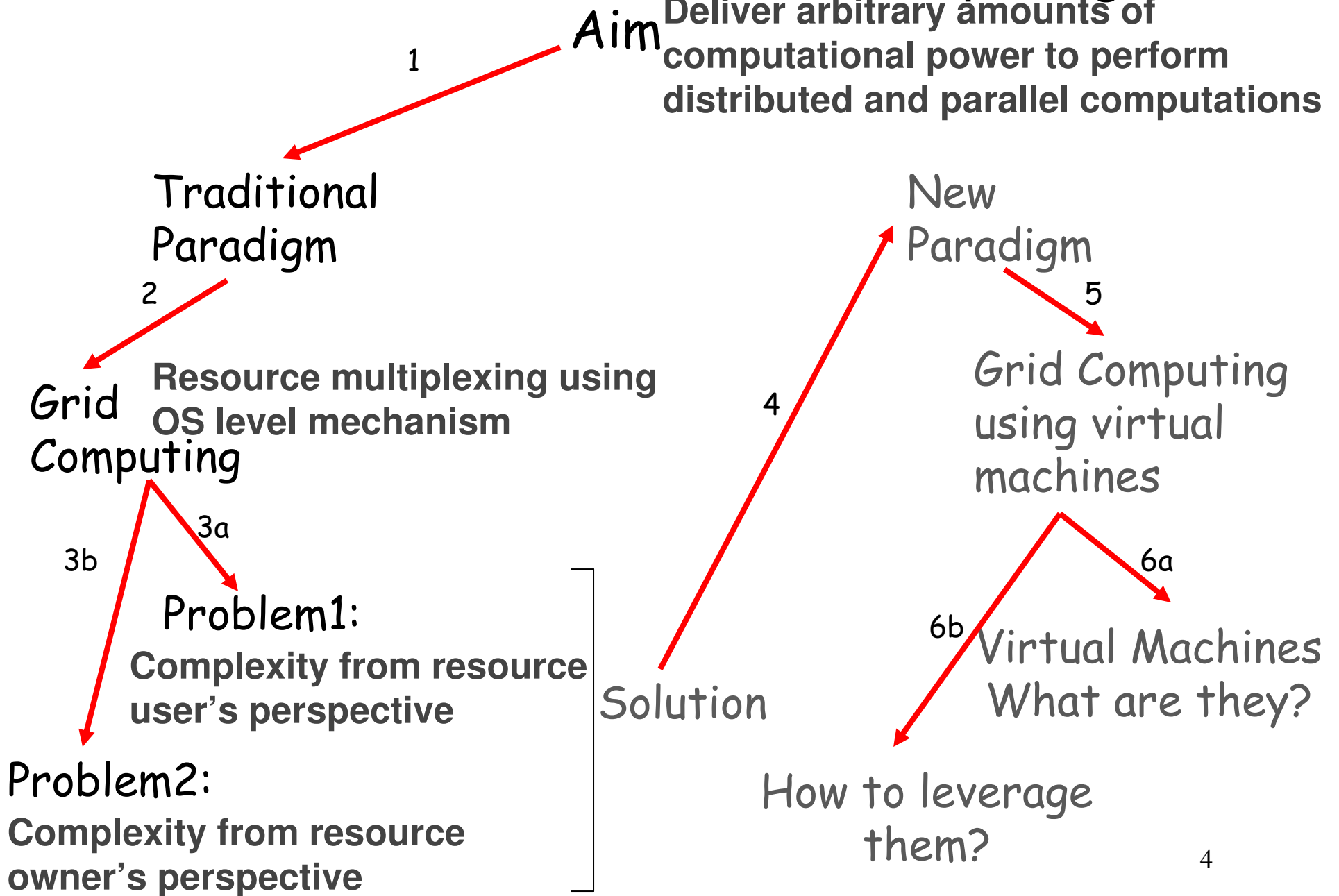
- Virtual execution environments provide opportunities for dynamic adaptation
- Important components are
 - Resource monitoring and inference
 - Application independent adaptation mechanisms
 - Efficient adaptation algorithms
- Previously proposed simple heuristics
 - Significant scope for improvement
- In this work
 - Formalize the adaptation problem
 - Show that it is NP-hard
 - Propose future research directions

Outline

- Virtual execution environments – An example
- Virtuoso system – Introduction
- Virtual networks
 - Measurement and inference (VTTIF)
 - Adaptation mechanisms (VNET)
- Problem formalization
- A special case of the adaptation problem
- Analysis
- Status
- Summary


Virtual Machine Grid Computing

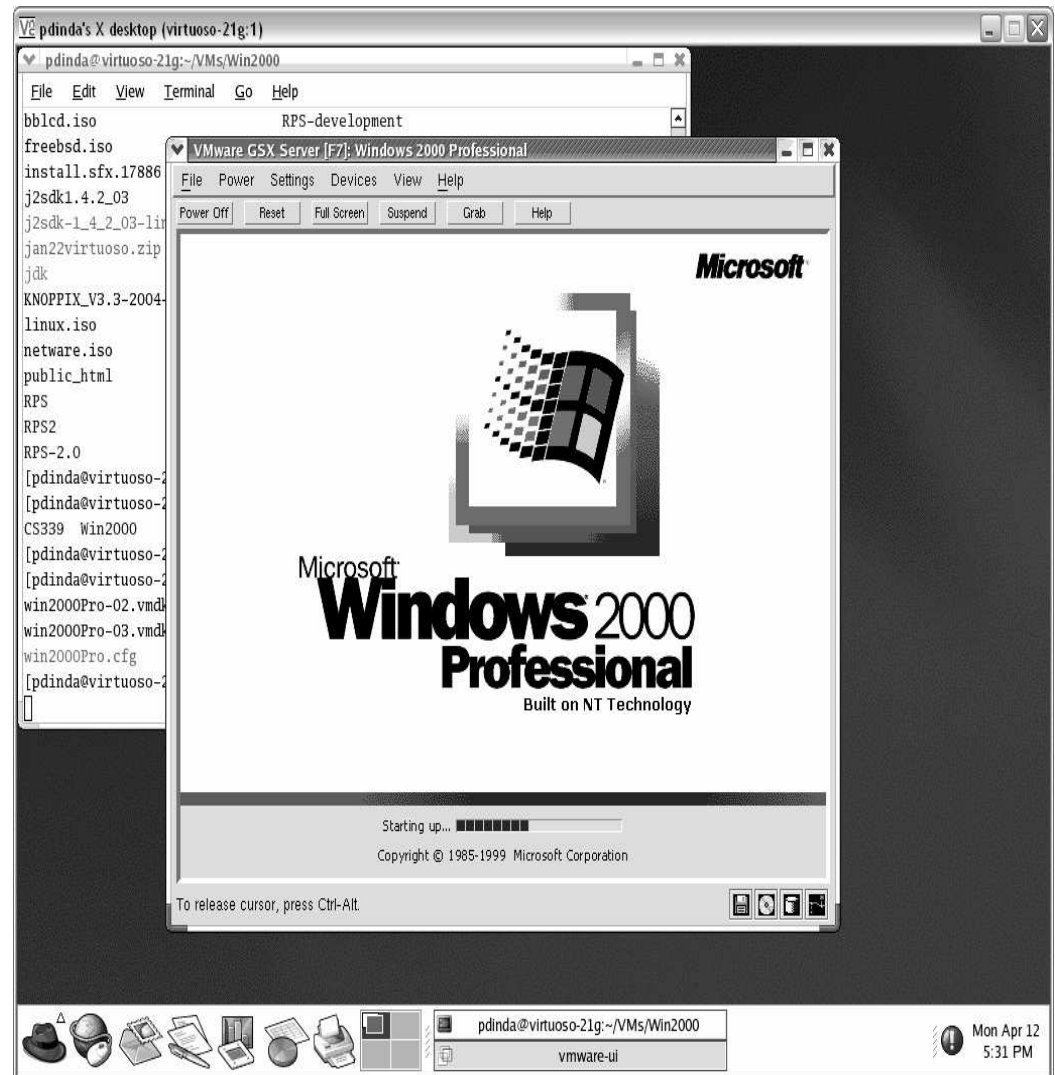
Aim Deliver arbitrary amounts of computational power to perform distributed and parallel computations



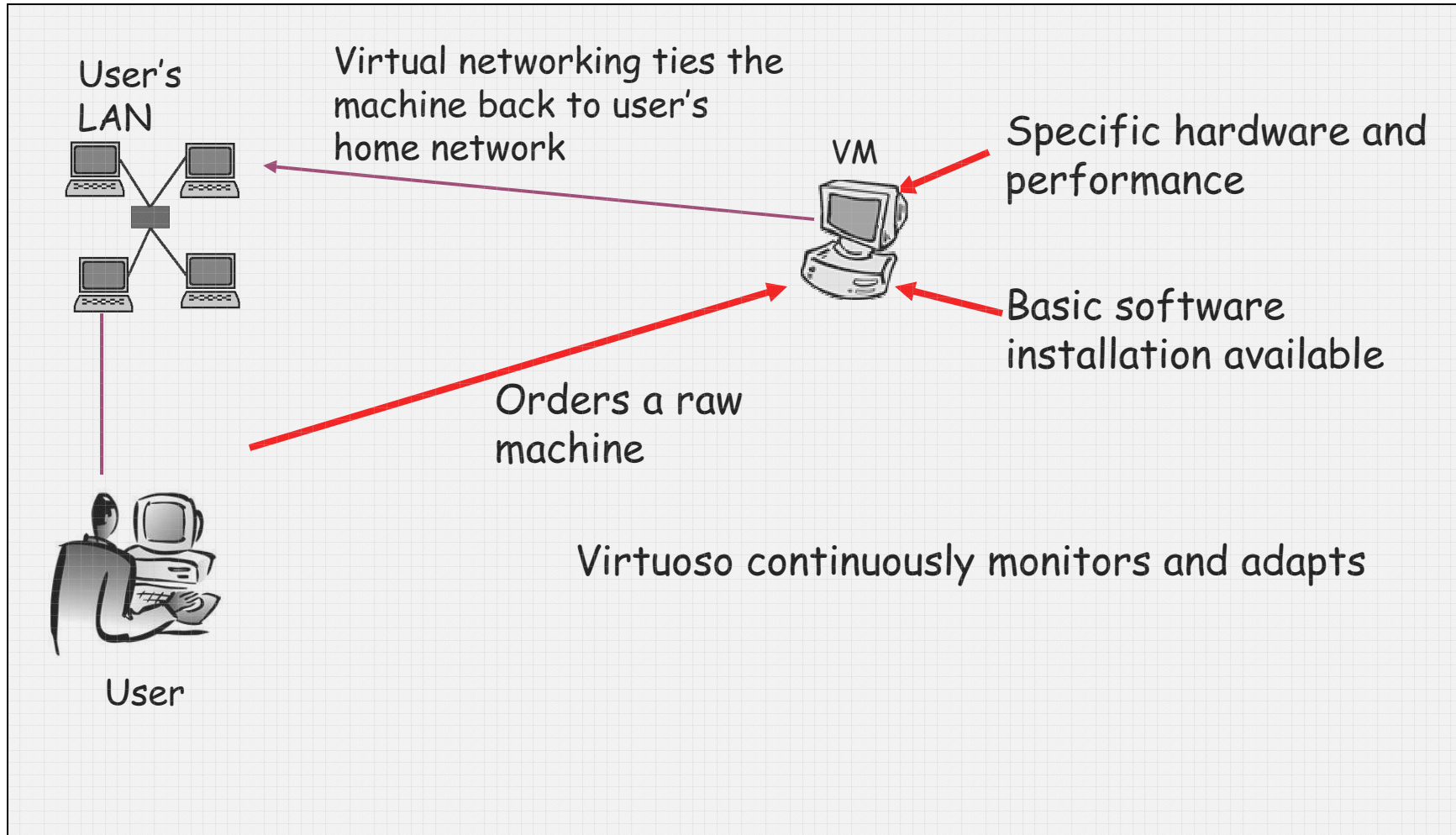
Virtual Machines

Virtual machine monitors (VMMs)

- Raw machine is the abstraction
- VM represented by a single image
- VMware GSX Server 



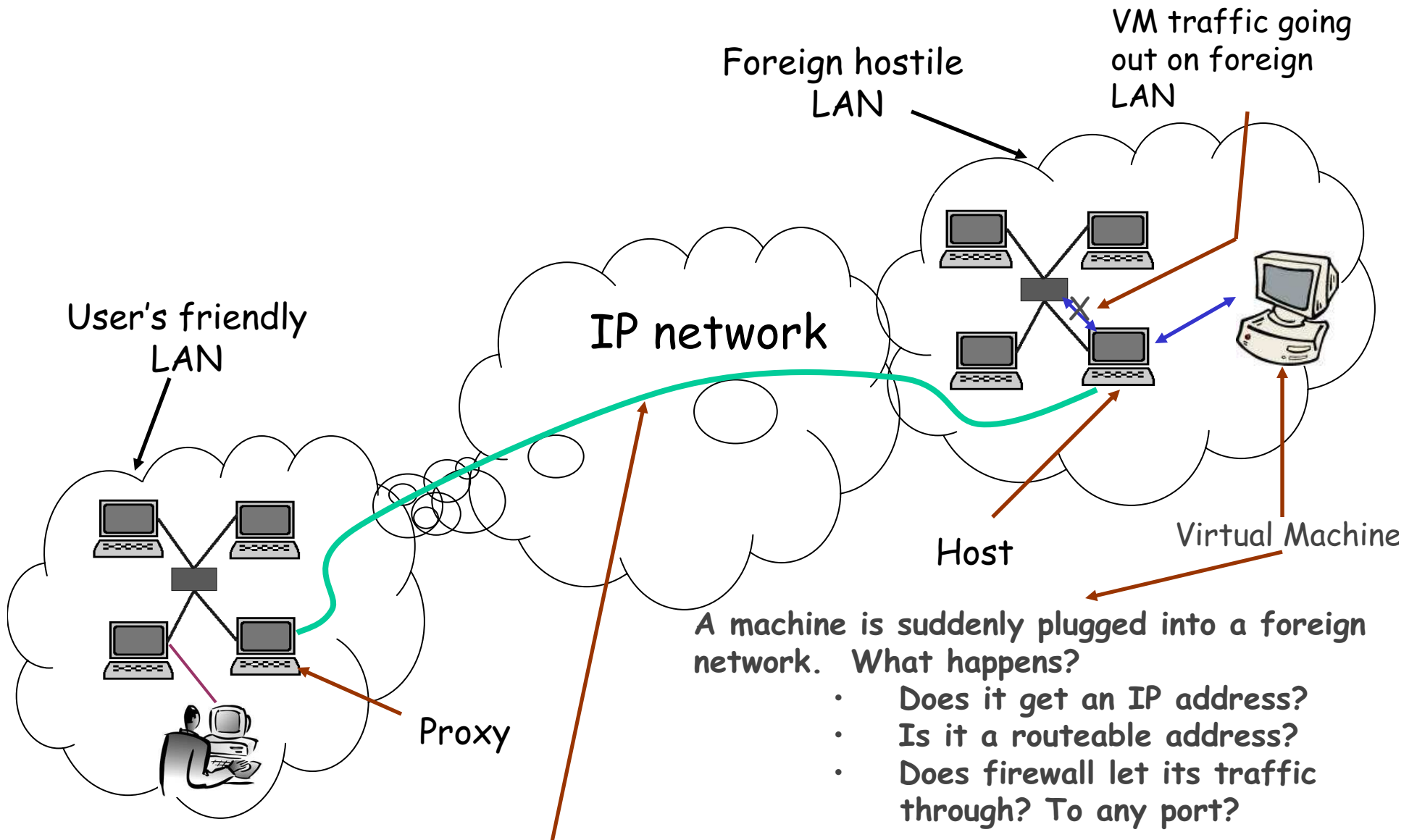
The Simplified Virtuoso Model



Outline

- Virtual execution environments – An example
- Virtuoso system – Introduction
- Virtual networks
 - Measurement and inference
 - Adaptation mechanisms
- Problem formalization
- A special case of the adaptation problem
- Analysis
- Status
- Summary

Virtual Networks



VNET: A bridge with long wires

Measurement and Inference

Underlying network

- Topology
- Bandwidth
- Latency

[Gupta et al. In submission]

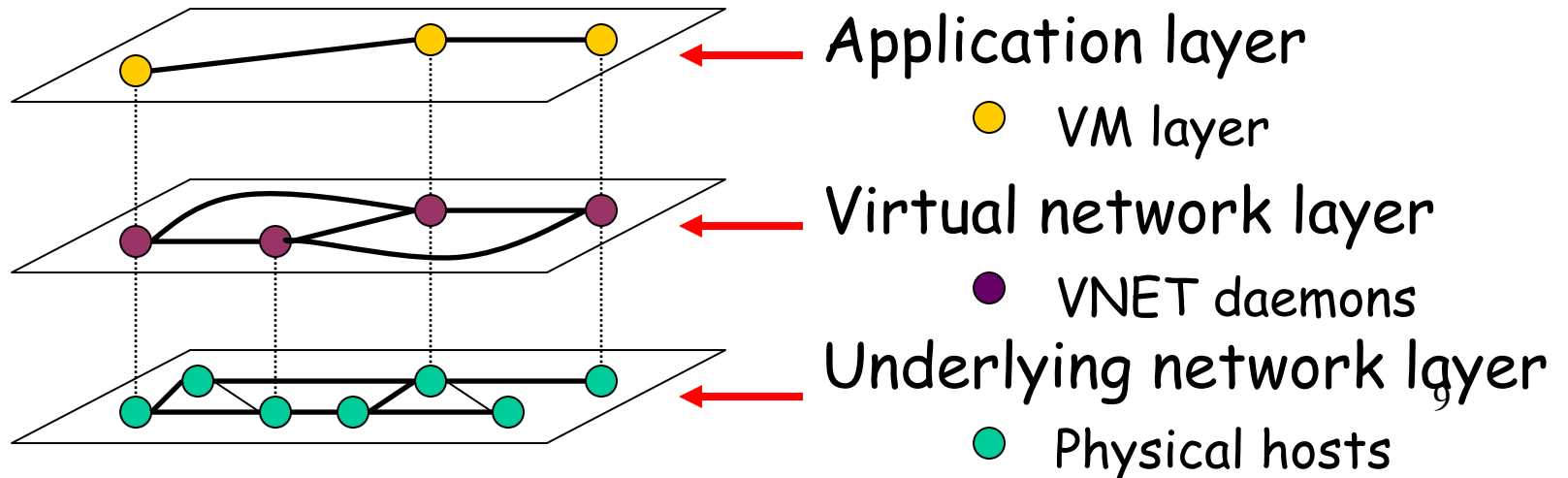
Host and VM

- Size and compute capacities
- Size and compute demands

Application (VTTIF)

- Topology
- Traffic load

[Gupta et al. LNCS 05]



Adaptation Mechanisms

VM Migration

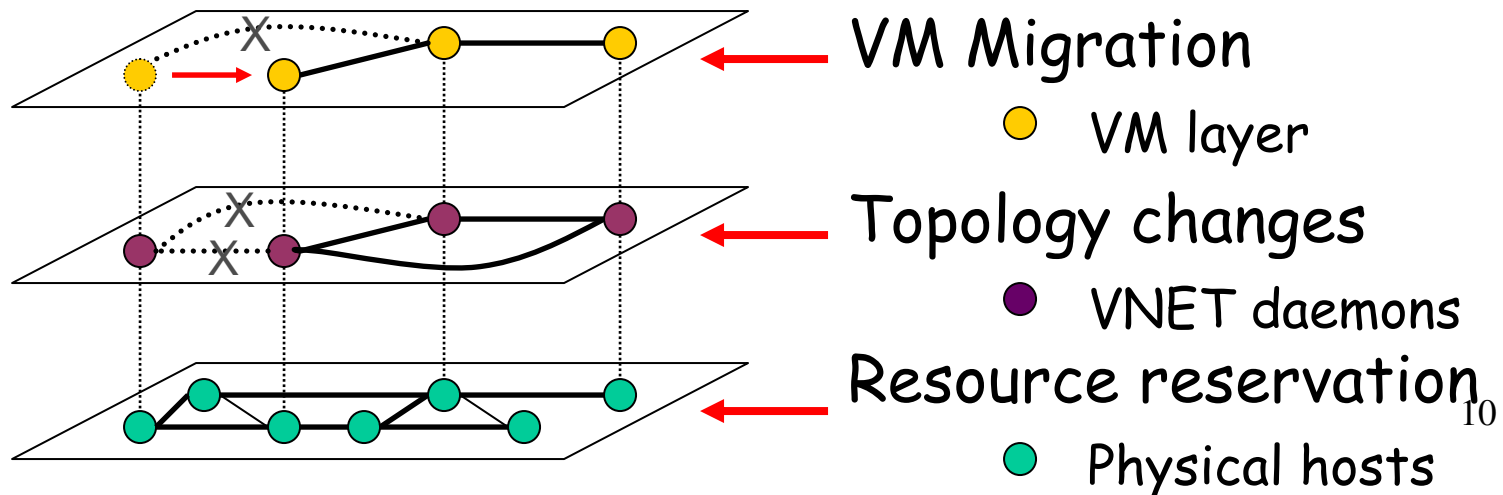
- Third party migration schemes

Topology changes

- Overlay links
 - Overlay forwarding rules
- [Sundararaj et al. LCR 04, HPDC 05]

Resource reservation

- Network [Lange et al. HPDC 05]
- CPU [Lin et al. GRID 2004]



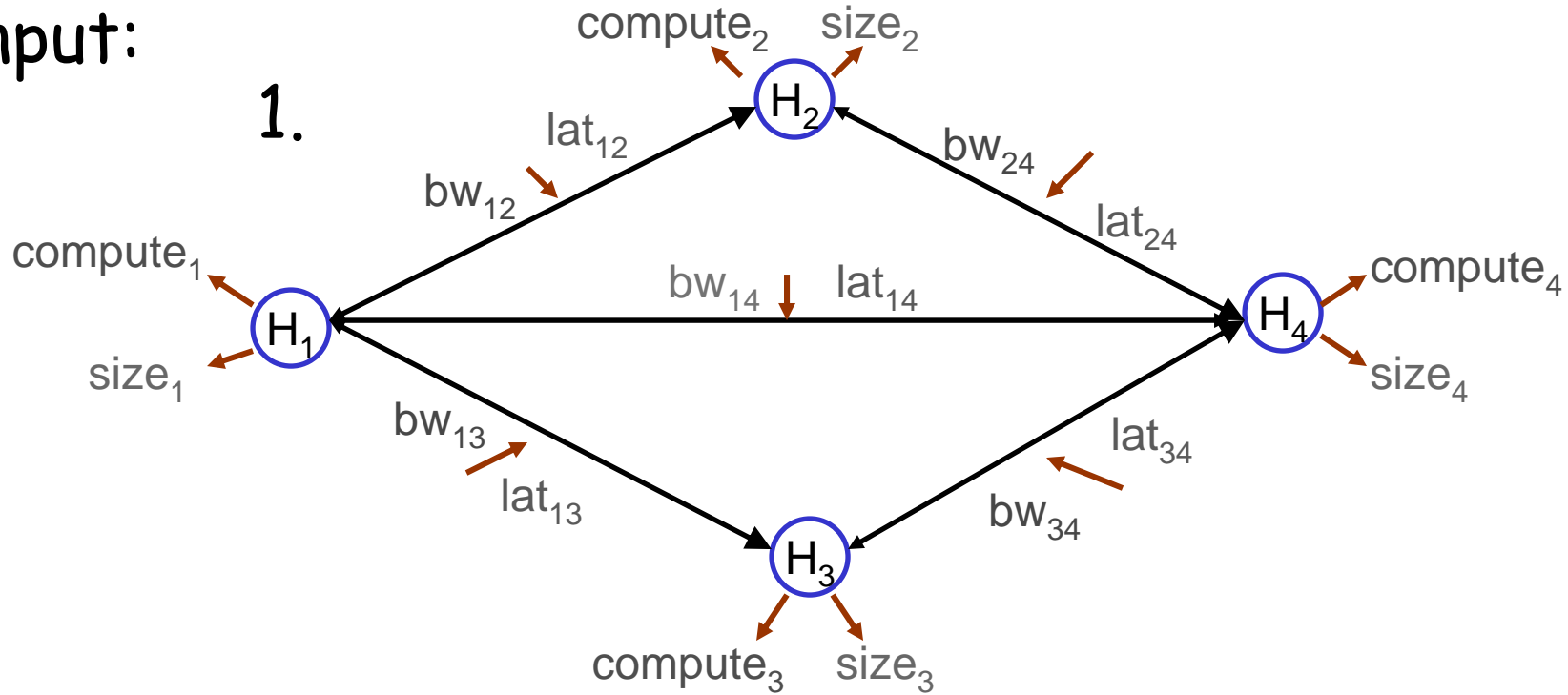
Outline

- Virtual execution environments – An example
- Virtuoso system – Introduction
- Virtual networks
 - Measurement and inference (VTTIF)
 - Adaptation mechanisms (VNET)
- **Problem formalization**
- **A special case of the adaptation problem**
- **Analysis**
- **Status**
- **Summary**

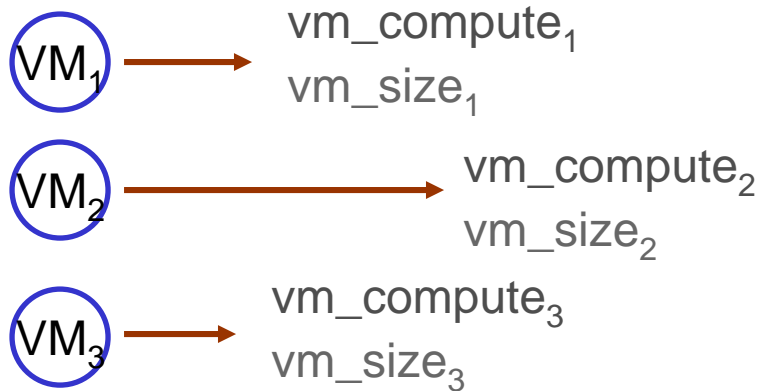
Generic Adaptation Problem In Virtual Execution Environments (GAPVEE)

Input:

1.



2.



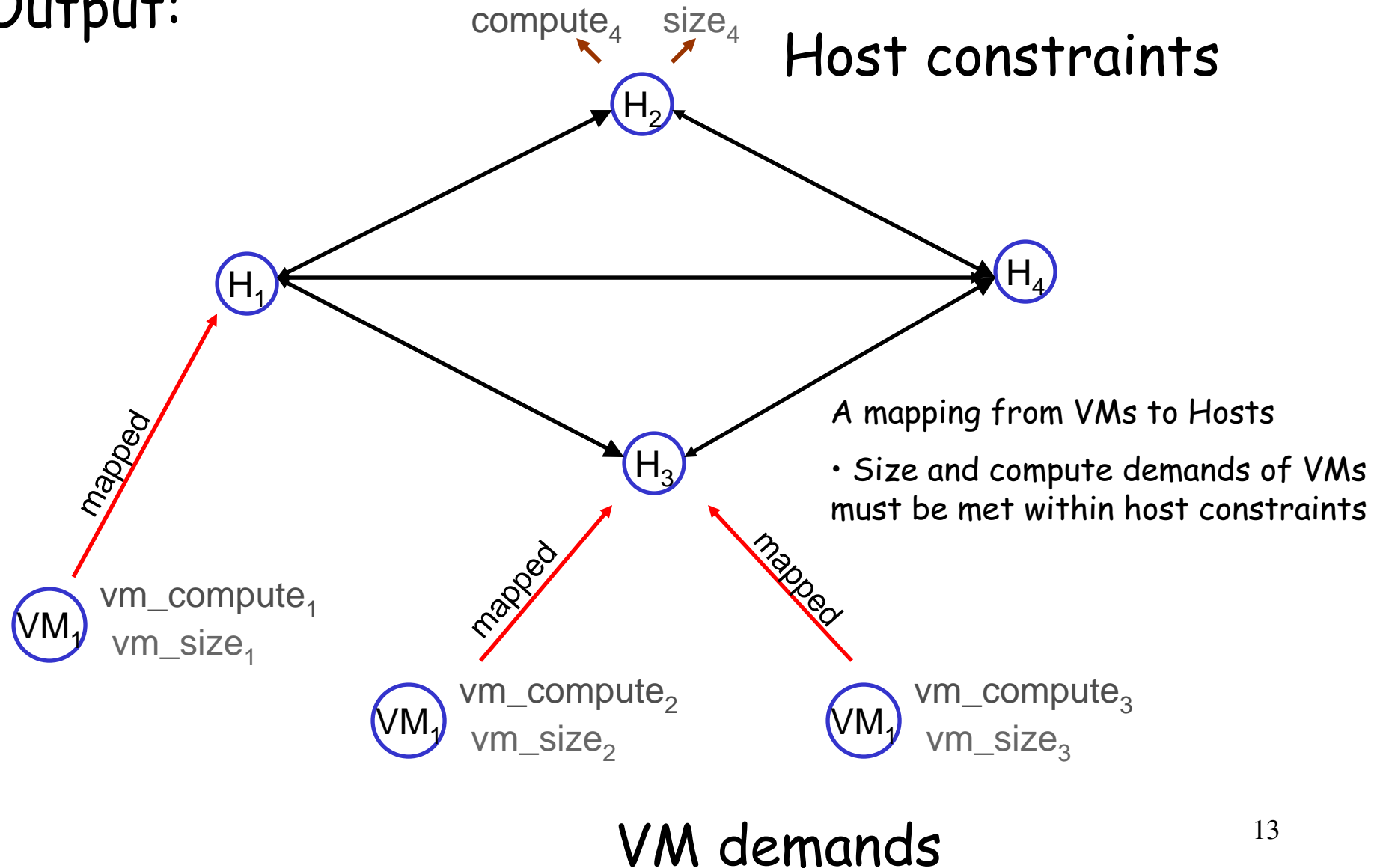
3.

	s_i	d_i	b_i	l_i
A_1	VM_1	VM_2	b_1	l_1
A_2	VM_1	VM_3	b_2	l_2
A_3	VM_2	VM_3	b_3	l_3
A_4	VM_3	VM_1	b_4	l_4

A set of ordered 4-tuples¹², "A"

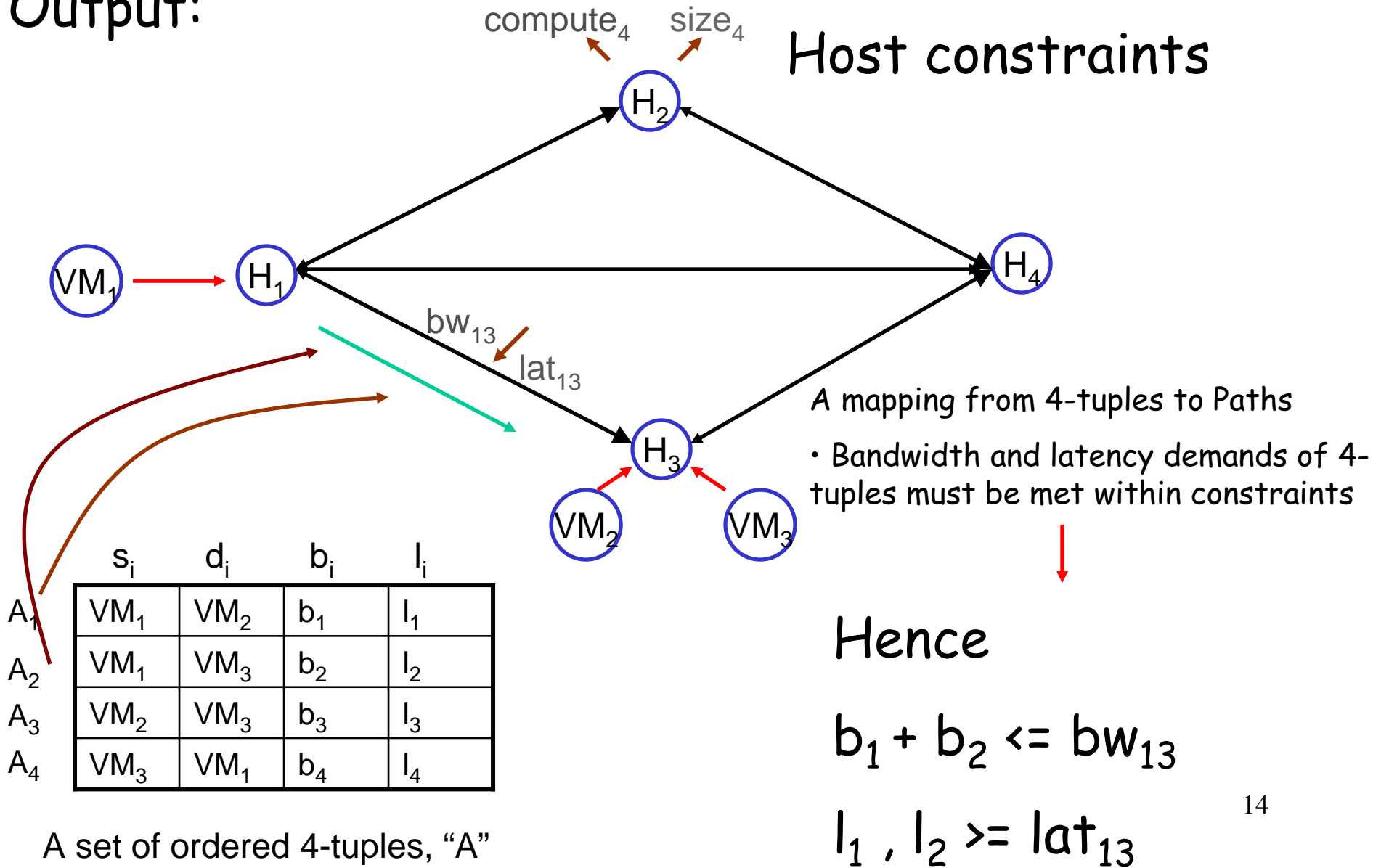
Generic Adaptation Problem In Virtual Execution Environments

Output:



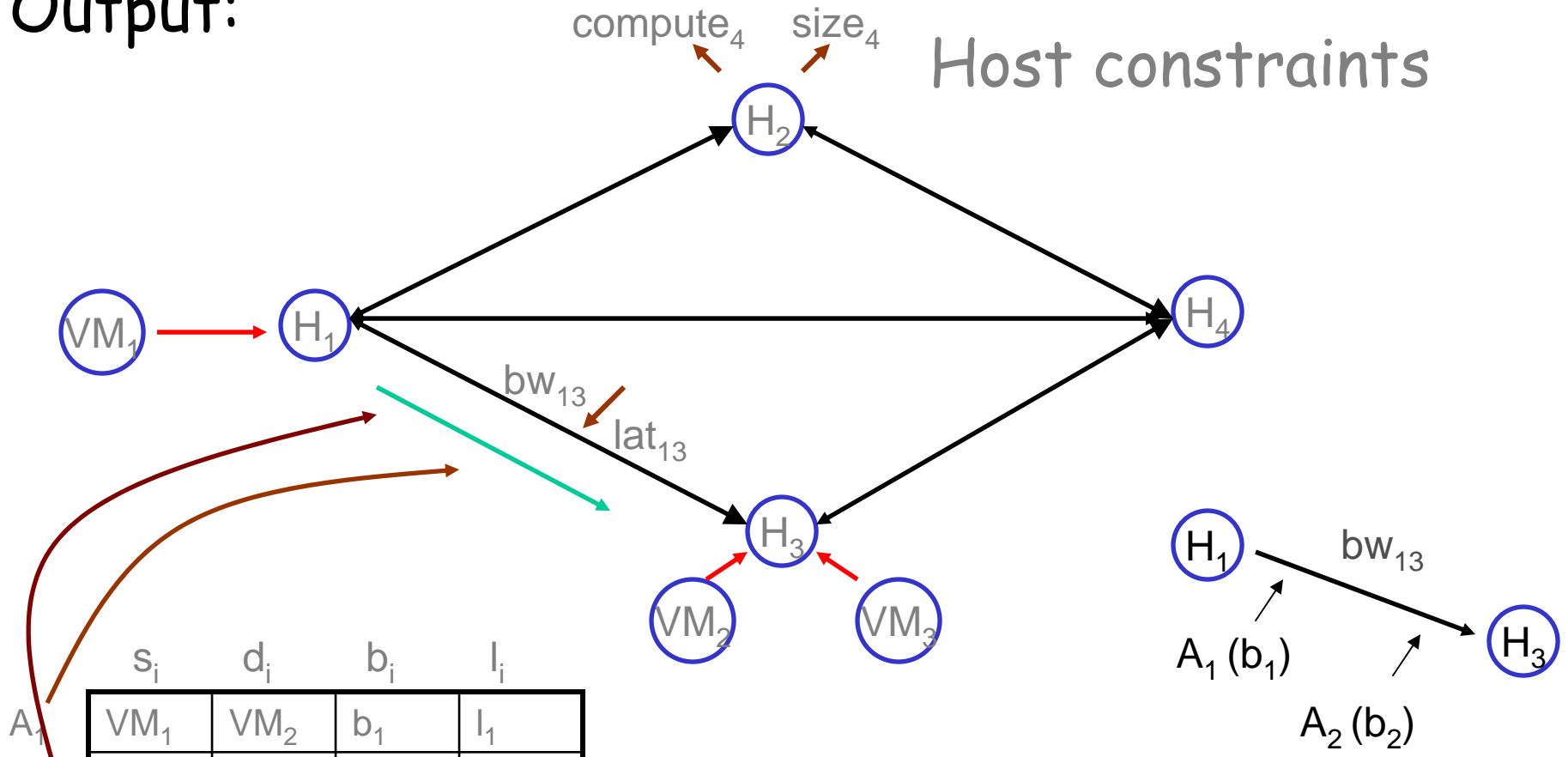
Generic Adaptation Problem In Virtual Execution Environments

Output:



Generic Adaptation Problem In Virtual Execution Environments

Output:



$bw_{13} - (b_1 + b_2)$ is the residual capacity of the edge

Bottleneck bandwidth: min residual capacity along a path

A set of ordered 4-tuples, "A"

Generic Adaptation Problem In Virtual Execution Environments

- Goal:
 - VMs to Hosts mapping
 - Path to each 4-tuple
 - Meeting all demands within constraints
 - Such that
 - Sum of residual bottleneck bandwidth over each mapped path is maximized

Optimizing Objective functions

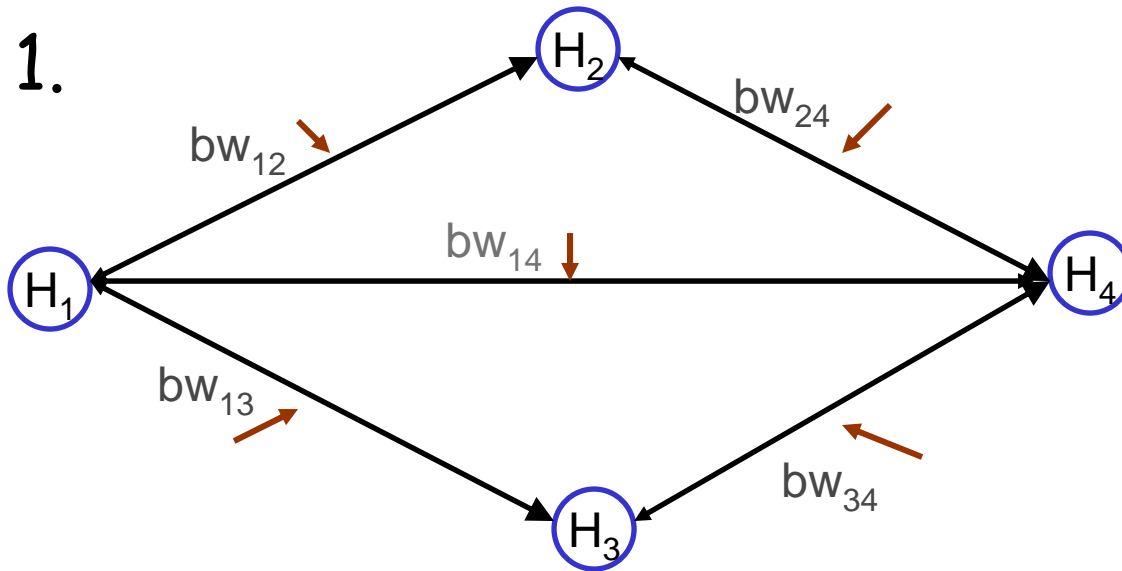
- Many possibilities
- Maximizing sum of residual bottleneck bandwidths over each mapped path
 - Intuition:
 - Leave the most room for application to increase performance
- Minimizing the residual bottleneck capacity
 - Intuition:
 - Increase room for other applications to enter system

Outline

- Virtual execution environments – An example
- Virtuoso system – Introduction
- Virtual networks
 - Measurement and inference (VTTIF)
 - Adaptation mechanisms (VNET)
- Problem formalization
- A special case of the adaptation problem
- Analysis
- Status
- Summary

Special Case of GAPVEE: Routing Problem In Virtual Execution Environments (RPVEE)

Input:



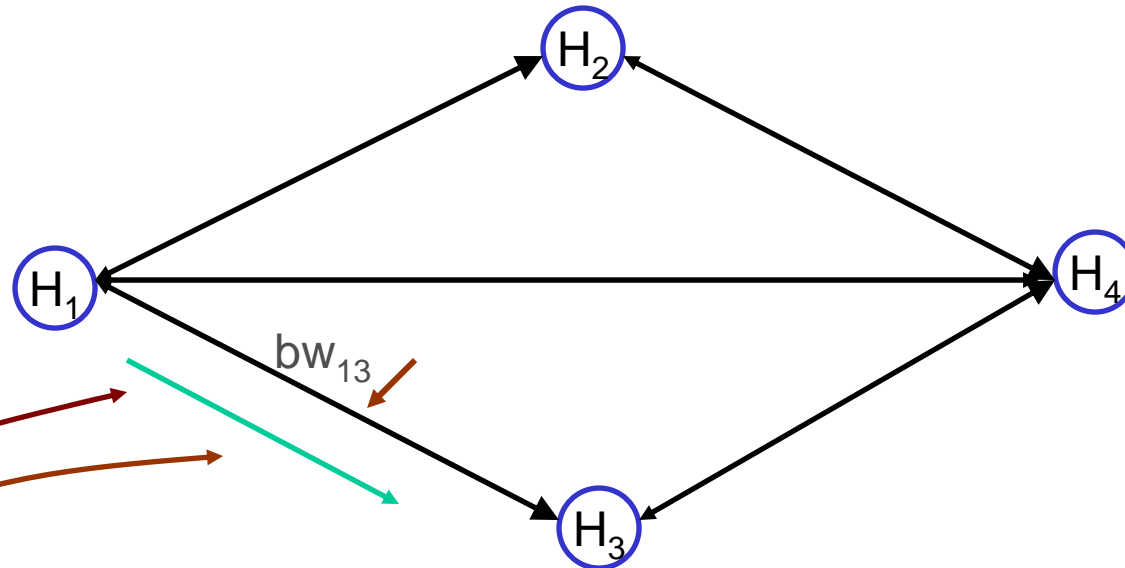
2.

	s_i	d_i	b_i
A_1	H_1	H_3	b_1
A_2	H_1	H_3	b_2
A_3	H_2	H_3	b_3
A_4	H_3	H_4	b_4

A set of ordered 3-tuples, "A"

Special Case of GAPVEE: Routing Problem In Virtual Execution Environments (RPVEE)

Output:



	s_i	d_i	b_i
A_1	H_1	H_3	b_1
A_2	H_1	H_3	b_2
A_3	H_2	H_3	b_3
A_4	H_3	H_4	b_4

A set of ordered 3-tuples, "A"

A mapping from 3-tuples to Paths

- Bandwidth demands of 3-tuples must be met within constraints
- Sum of residual bottleneck bandwidth is maximized over each path

Analysis

- Theorem 1: RPVEE is NP-hard
- Edge Disjoint Path Problem (EDPP)
 - Arbitrary instance of EDPP
 - Convert to an instance of RPVEED
 - A “Yes” solution for RPVEED implies a “Yes” solution for EDPP
 - A “No” solution for RPVEED implies a “No” solution for EDPP

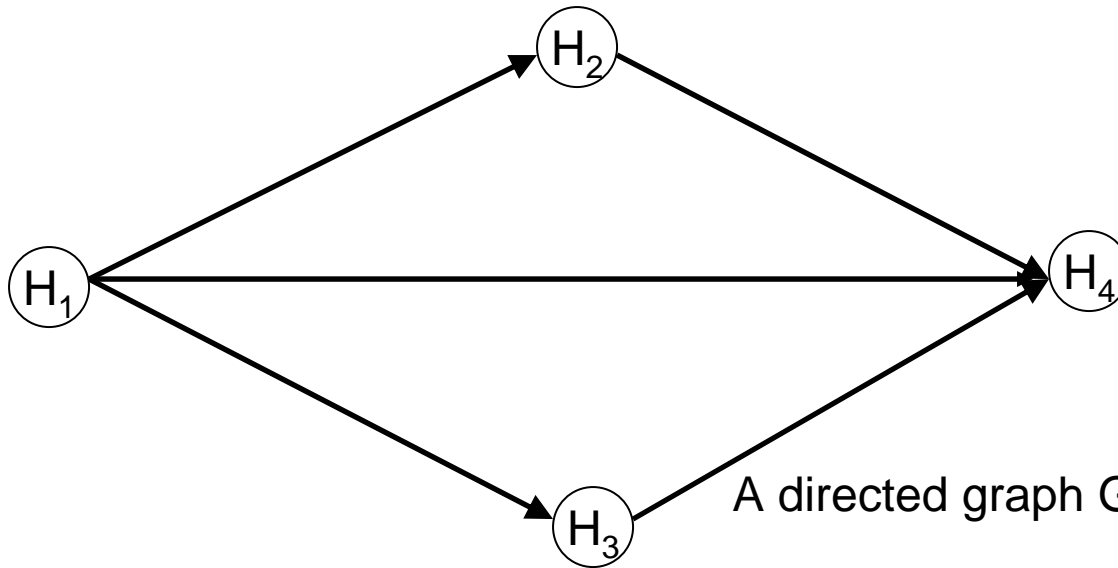
Edge Disjoint Path Problem (EDPP)

- A well known NP-complete problem
- Input
 - A graph $G = (H, E)$
 - A set of ordered 2-tuples $S = \{(s_i, d_i), \text{ where } s_i, d_i \text{ in } H\}$
- Output
 - “Yes” if for all (s_i, d_i) in S , there exist edge disjoint paths from s_i to d_i in $G = (H, E)$
 - “No” otherwise

Decision version of RPVEE (RPVEED)

- Input
 - A graph $G = (H, E)$
 - A function $bw: E \rightarrow R$
 - A set of ordered 3-tuples $S = \{(s_i, d_i, b_i), \text{ where } s_i, d_i \text{ in } H, b_i \text{ in } R, i = 1, \dots, k\}$
- Output
 - “Yes” if for all (s_i, d_i, b_i) in S , there exist paths from s_i to d_i in $G = (H, E)$, such that sum of bottleneck bandwidth = $k \cdot \varepsilon$
 - “No” otherwise

Given an arbitrary instance of EDPP

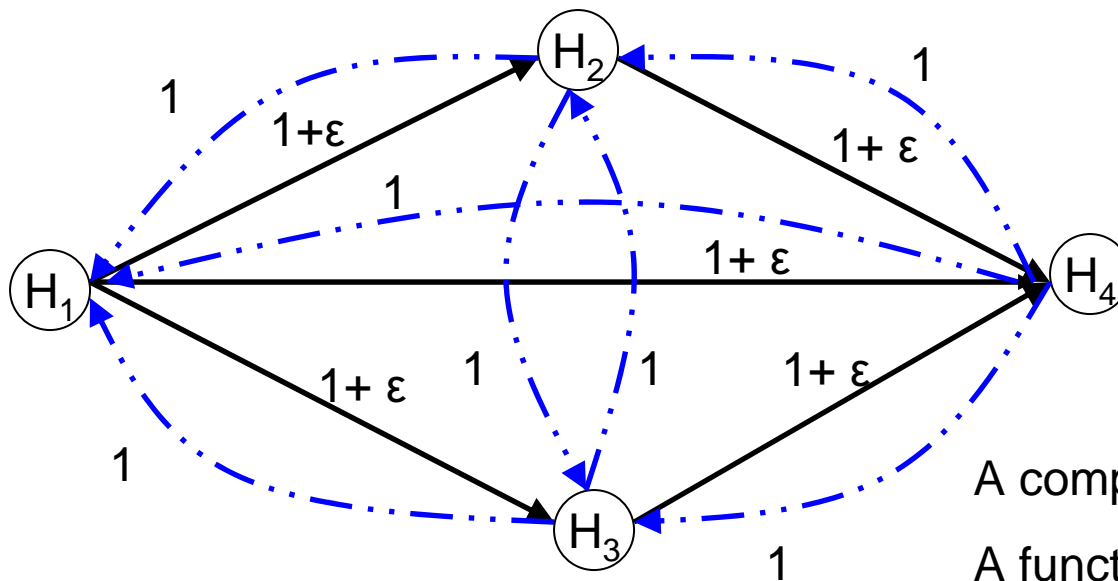


s_i	d_i
H_1	H_2
H_1	H_4
H_2	H_4
H_1	H_3

A set of ordered 2-tuples

A directed graph $G = (H, E)$

Converted to a particular instance of RPVEED



s_i	d_i	b_i
H_1	H_2	1
H_1	H_4	1
H_2	H_4	1
H_1	H_3	1

A set of ordered 3-tuples

A complete directed graph $G_{24} = (H, E)$

A function $bw : E \rightarrow \mathbb{R}$

Outline

- Virtual execution environments – An example
- Virtuoso system – Introduction
- Virtual networks
 - Measurement and inference (VTTIF)
 - Adaptation mechanisms (VNET)
- Problem formalization
- A special case of the adaptation problem
- Analysis
- **Status**
- **Summary**

Status

- Previously developed a variety of heuristics (Greedy and Simulated annealing)
- Effective in improving performance
 - Significant scope for improvement
- Formalization and analysis, first steps
- Generic incarnation hard, focus on special cases
- Currently researching the well studied variants of our problem (such as un-splittable flows)

Summary

- Virtual execution environments provide opportunities for dynamic adaptation
- Important components are
 - Resource monitoring and inference
 - Application independent adaptation mechanisms
 - Efficient adaptation algorithms
- Previously proposed simple heuristics
 - Significant scope for improvement
- In this work
 - Formalize the adaptation problem
 - Show that it is NP-hard
 - Propose future research directions



- For More Information
 - Prescience Lab (Northwestern University)
 - <http://plab.cs.northwestern.edu>
 - Virtuoso: Resource Management and Prediction for Distributed Computing using Virtual Machines
 - <http://virtuoso.cs.northwestern.edu>
- VNET is publicly available from
 - <http://virtuoso.cs.northwestern.edu>