Virtual Clusters and Resource Management : Migration of Memory, Files and Network Resources & Dynamic Deployment of Clusters

Y. V. Lokeswari AP/ CSE

SSN College of Engineering

Reference: Distributed and Cloud Computing
K. Hwang, G. Fox and J. Dongarra

Design Issues of Virtual Cluster

- Live Migration Of VMs
- Memory, File system and Network Migration
- Dynamic Deployment of Virtual Clusters

Outline

- Memory Migration
- File System Migration
- Network Migration
- Live Migrations of VMs between 2 Xen enabled Hosts.
- Dynamic Deployment of Virtual Clusters

Memory Migration

- **Memory Migration** is in the range of hundreds of megabytes to few gigabytes.
- In **Internet Suspend-Resume** (ISR) technique memory states are likely to **overlap** in suspended and resumed instances of VM.
- ISR exploits **Temporal Locality**
- Temporal Locality refers that memory states differ **only** by the **amount** of **work** done **since VM** is last **suspended** and before being **initiating migration**.
- Each **file** in a **file system** must be represented as a **tree** of **small sub-files**.
- The **copy** of tree **exist** in both **suspended** and **resumed** VM
- This ensures that **transmission only** for the **files** which have been **changed**.

File System Migration

- In **VM migration**, system should provide VM with **consistent** and **location independent** view of **file system** available on all host.
- Provide each VM with its **own virtual disk** which the file system is mapped.
- Now **transport** the contents of **Virtual disk** along with other **VM states**.
- If the **disk capacity** is very **high**, maintain **global file system** across **all machines** where **VM** can be **located**.
- Instead of having a **distributed file system**, every VM will have access only to its **local file system**.
- **Copy** the **virtual disk** contents to VM's **local file system** for resumed VM.
- **Spatial Locality**: Transmit only **difference** between **two file systems** at suspending and resuming locations

Network Migration

- **Migrating VM** should maintain all **open connections without** relying on **forwarding** or **redirection** mechanisms.
- To **locate** and **communicate** with VM, each **VM** is assigned with **Virtual IP** address, **distinct** from **IP** address of **host**.
- **VMM** maintains **mapping** of **Virtual IP** with **Virtual MAC address**.
- If **source** and **destination** machines of VM migration are in a **single LAN**, migrating host **advertise** that **IP** is moved to **new location**, which reconfigure all **peers** to **send** files to **new location**.

Copy Strategies in Live Migration

- **Live migration** means moving **VM** from one physical machine to another, while **keeping OS** environment and **applications unbroken**.
- **Pre-copy**: VM could be migrated **without suspending** VM and keep **applications running** during migration.
- **First transfer** copies **all** the **pages** and copies **only modified** memory pages on subsequent **iterations** until **writeable** working set becomes **small**..
- **Drawback**: This consumes **large** amount of **network bandwidth** to transfer dirty pages in each round.
- When the network bandwidth is limited, **total migration** time **increases** to 10 minutes.
- **Solution**: We can set **maximum number** of **iterations**.

Copy Strategies in Live Migration

- A **Checkpointing/recovery** and **trace/replay** approach is proposed to enable **fast VM migration**. This reduces downtime and total migration time.
- **Post-Copy**: All **memory pages** are transferred **only once** during the whole **migration** process.
- **Advantage**: Total **migration time** is **reduced**.
- **Drawback**: **Downtime** is **higher** because latency of **fetching pages** from source node before VM is resumed on target.
- **Solution**: Memory **compression** and **decompression** algorithms can be applied to **reduce memory overhead**.

Live Migration of VM in Xen environment

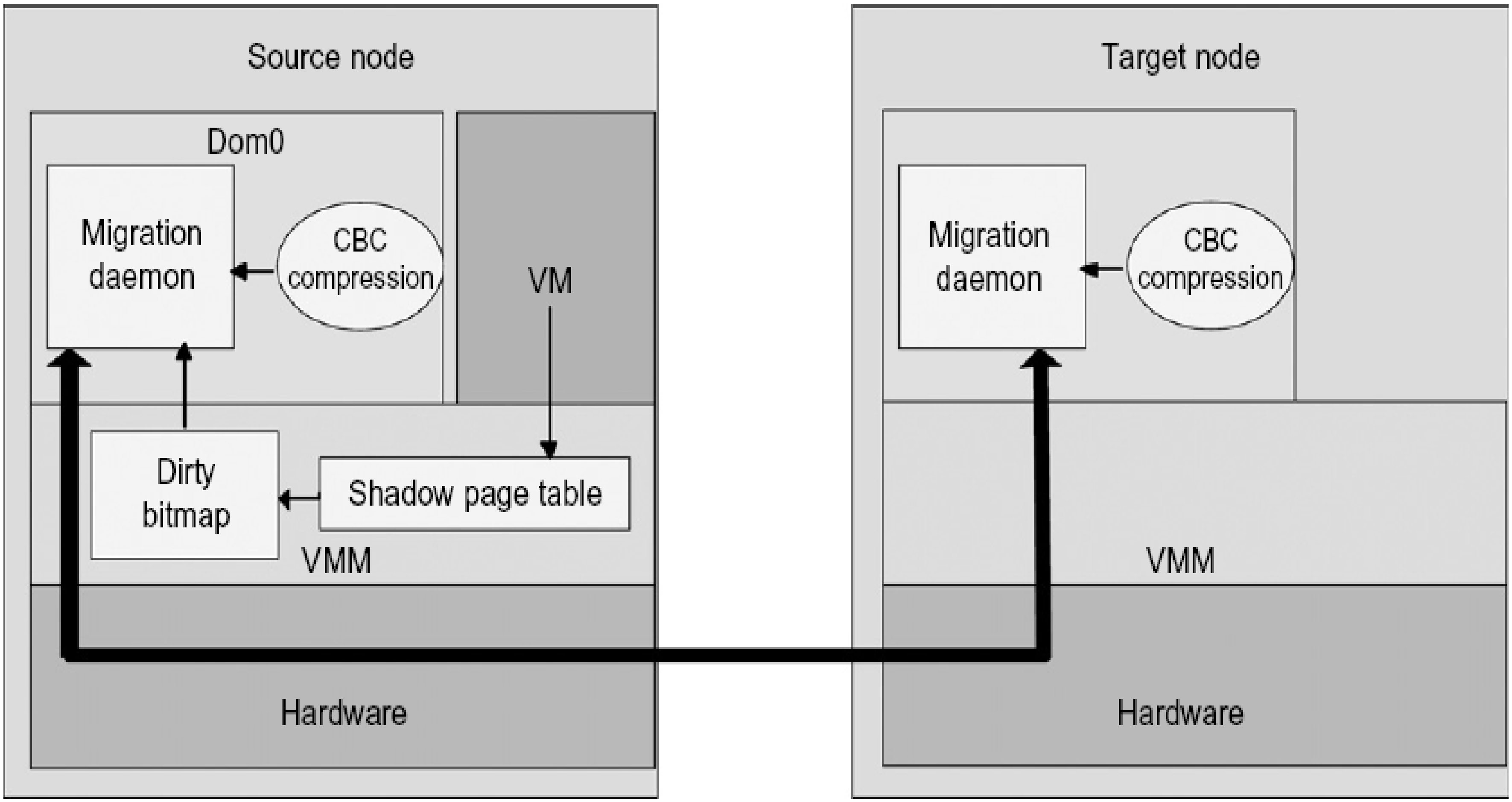

FIGURE 3.22

Live migration of VM from the Dom0 domain to a Xen-enabled target host.

Dynamic Deployment of Virtual Clusters

Virtual Cluster Projects

Table 3.5 Experimental Results on Four Research Virtual Clusters

Project Name	Design Objectives	Reported Results and References
Cluster-on-Demand at Duke Univ.	Dynamic resource allocation with a virtual cluster management system	Sharing of VMs by multiple virtual clusters using Sun GridEngine [12]
Cellular Disco at Stanford Univ.	To deploy a virtual cluster on a shared-memory multiprocessor	VMs deployed on multiple processors under a VMM called Cellular Disco [8]
VIOLIN at Purdue Univ.	Multiple VM clustering to prove the advantage of dynamic adaptation	Reduce execution time of applications running VIOLIN with adaptation [25,55]
GRAAL Project at INRIA in France	Performance of parallel algorithms in Xen-enabled virtual clusters	75% of max. performance achieved with 30% resource slacks over VM clusters

COD Project @ DUKE Univ

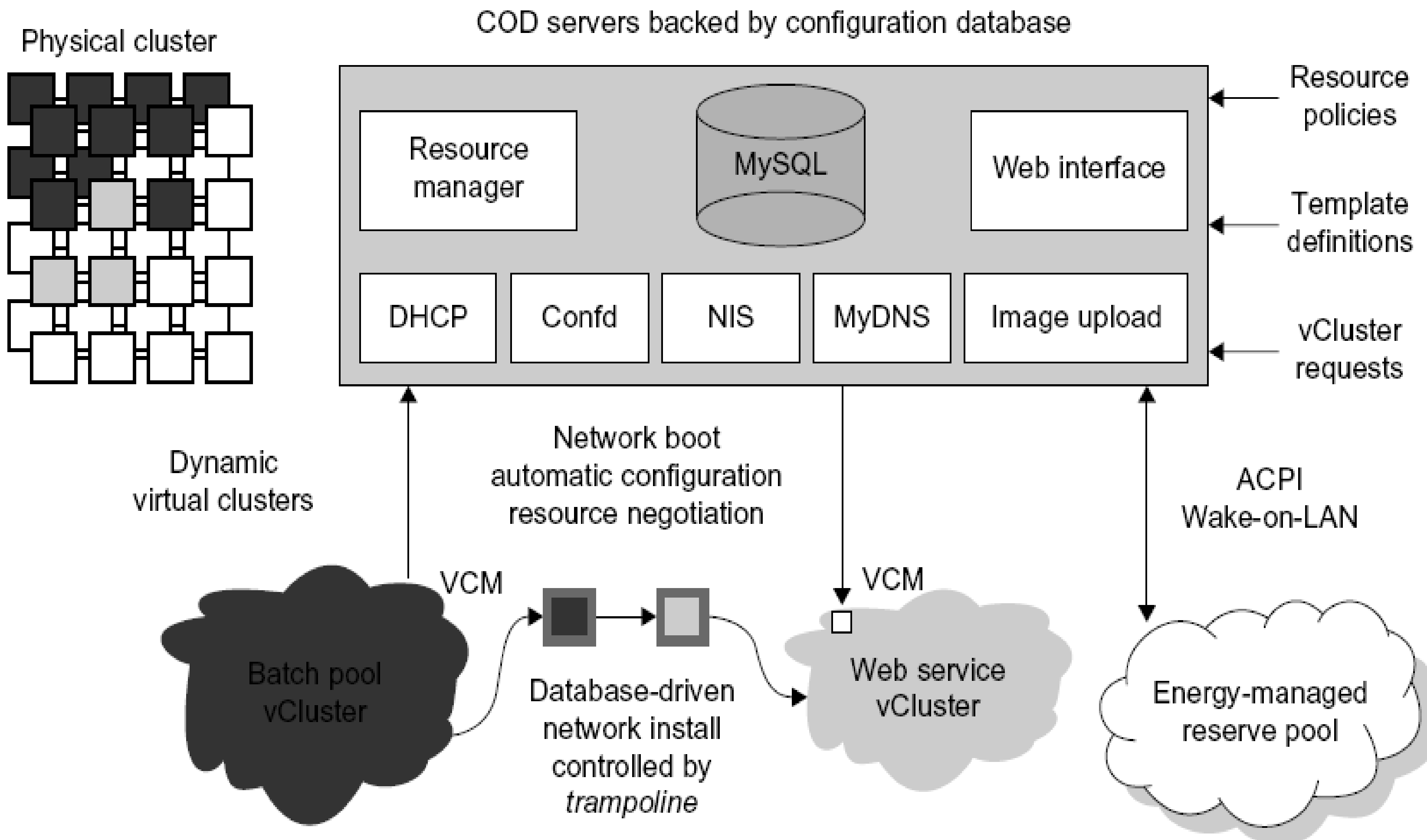

FIGURE 3.23

COD partitioning a physical cluster into multiple virtual clusters.

(Courtesy of Jeff Chase, et al, HPDC-2003 [12])

Cluster-on-Demand (COD Project) at Duke University

- The COD (**Cluster-on-Demand**) project is a **virtual cluster** management system for **dynamic allocation** of servers from a computing pool **to multiple virtual clusters**.
- The vClusters run a **batch schedule** from **Sun's GridEngine** on a web server cluster.
- The COD system can respond to **load changes** in **restructuring** the **virtual clusters dynamically**.
- The Duke researchers used the **Sun GridEngine scheduler** to demonstrate that **dynamic virtual clusters** are an enabling **abstraction** for advanced resource management in computing utilities such as grids.

Cluster-on-Demand (COD Project) at Duke University

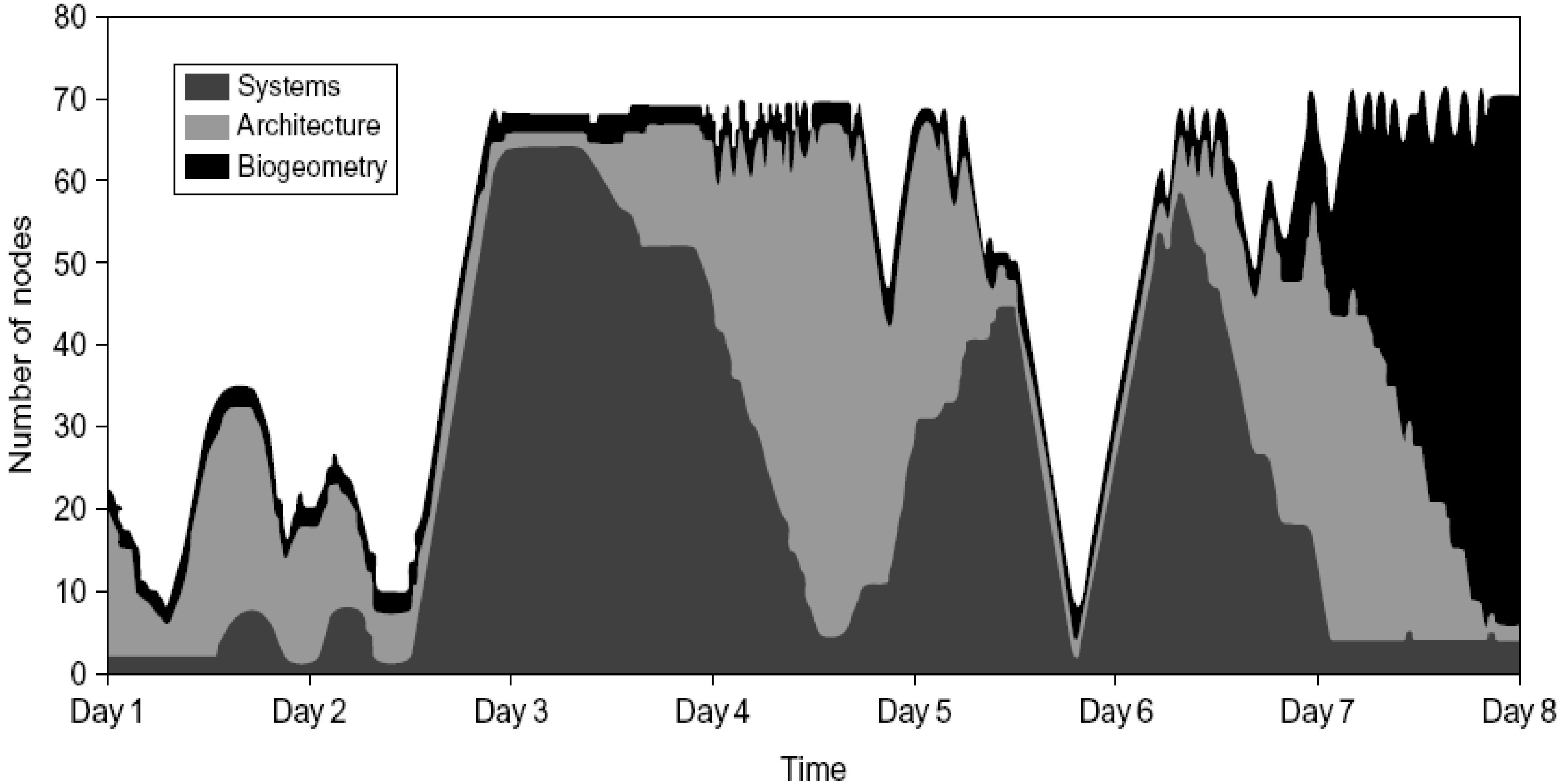

FIGURE 3.24

Cluster size variations in COD over eight days at Duke University.

(Courtesy of J. Chase, et al. [12])

VIOLIN Project at Purdue University

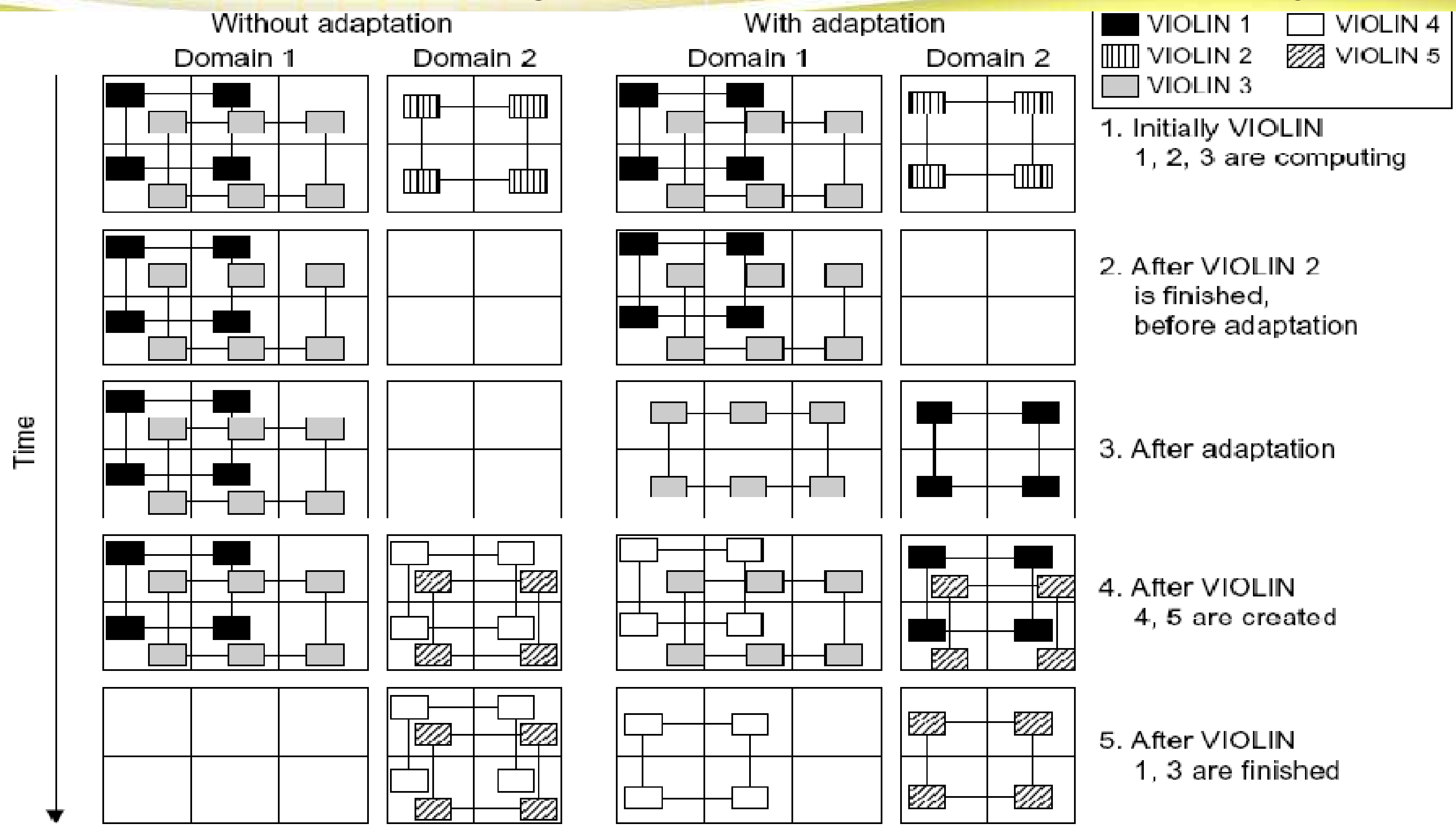

FIGURE 3.25

VIOLIN adaptation scenario of five virtual environments sharing two hosted clusters; Note that there are more idle squares (blank nodes) before and after the adaptation.

(Courtesy of P. Ruth, et al. [24,51])

VIOLIN Project at Purdue University

- The **Purdue VIOLIN** Project applies live **VM migration** to **reconfigure** a **virtual cluster** environment.
- Its purpose is to achieve **better resource utilization** in executing multiple **cluster jobs** on multiple **cluster domains**
- A **virtual execution** environment is able to **relocate** itself across the **infrastructure**.
- The adaptation is **transparent** to both users of virtual environments and administrations of infrastructures.

Summary

- Memory Migration
- File System Migration
- Network Migration
- Live Migrations of VMs between 2 Xen enabled Hosts.
- Dynamic Deployment of Virtual Clusters

Thank You