

Virtualization and Databases: State of the Art and Research Challenges

Ashraf Aboulnaga
University of Waterloo

Cristiana Amza
University of Toronto

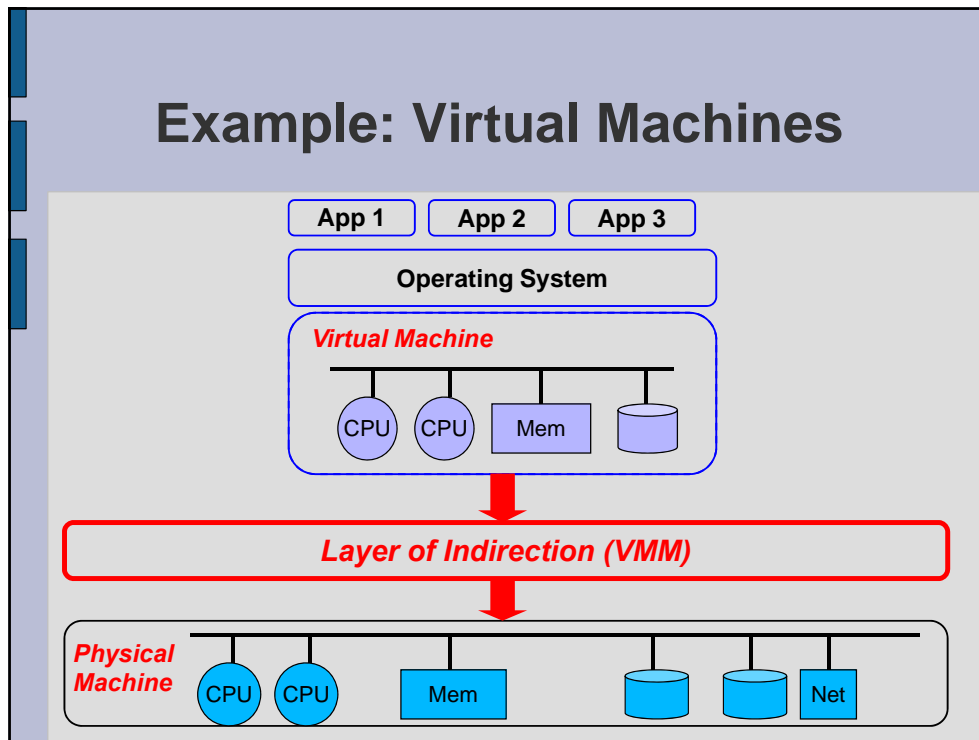
Kenneth Salem
University of Waterloo

What is Virtualization?

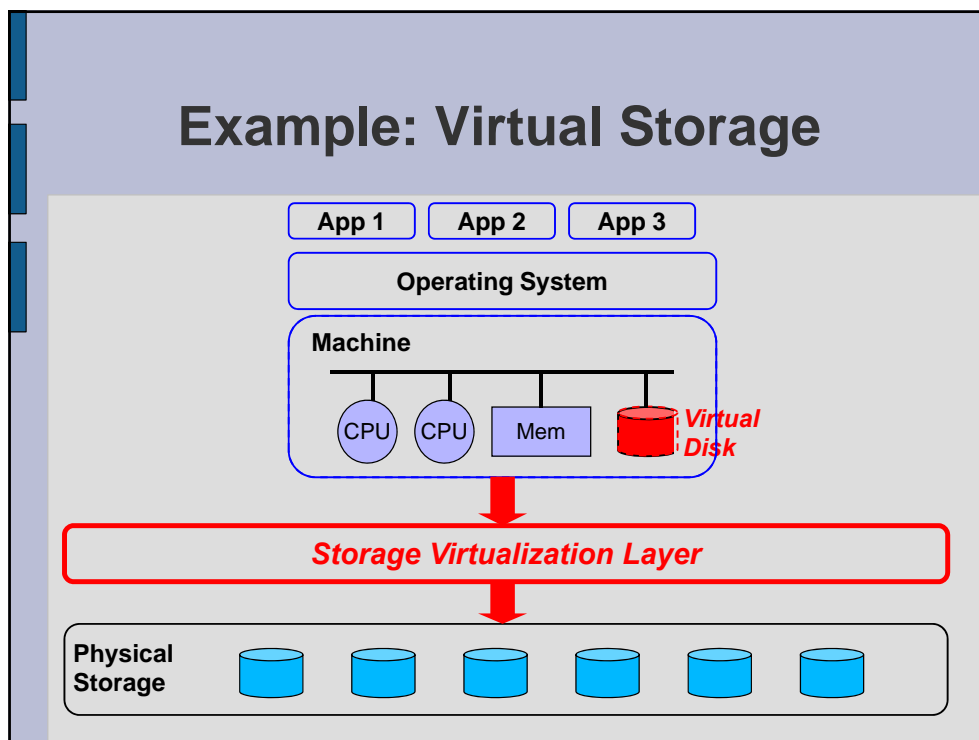
- Separating the abstract view of a computing resource or service from the implementation of this resource or service
- ***A layer of indirection between abstract view and implementation***
 - Hides implementation details
 - Controls mapping from abstract view to implementation

*"any problem in computer science can be
solved with another layer of indirection"*
– David Wheeler

Example: Virtual Machines



Example: Virtual Storage



Why Virtualization?

- Virtualization adds **flexibility** and **agility** to the computing infrastructure
- Can be used to solve many problems related to provisioning, manageability, security, ...
 - Pool and share computing resources
 - Simplify administration and management
 - Improve fault tolerance
- For organizations: **Lower total cost of ownership** for computing infrastructure
 - Fewer computing resources
 - More resilient and simpler to manage

Why Should We Care?

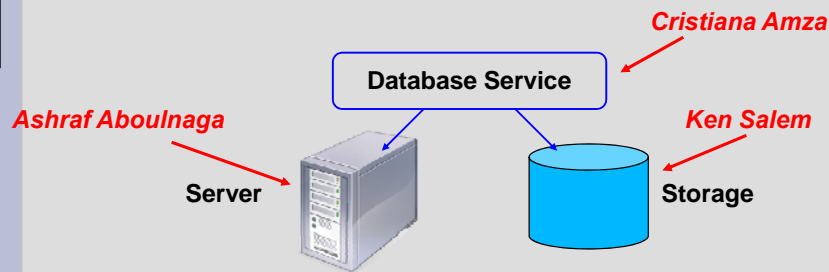
- Computing infrastructure is becoming more and more virtualized
- Database systems are increasingly being run in virtualized environments
- Does this introduce new opportunities or challenges for database systems?

YES!

"virtualization will be a \$20 billion market by 2010"
– IDC, January 2007

This Tutorial

- Virtualizing computing **resources** and **services**



- The term virtualization is also used in other areas

- ~~– Virtual teams~~
- ~~– Virtual enterprises~~
- ~~– Java virtual machines~~
- ~~– Virtual reality~~
- ~~– ...~~

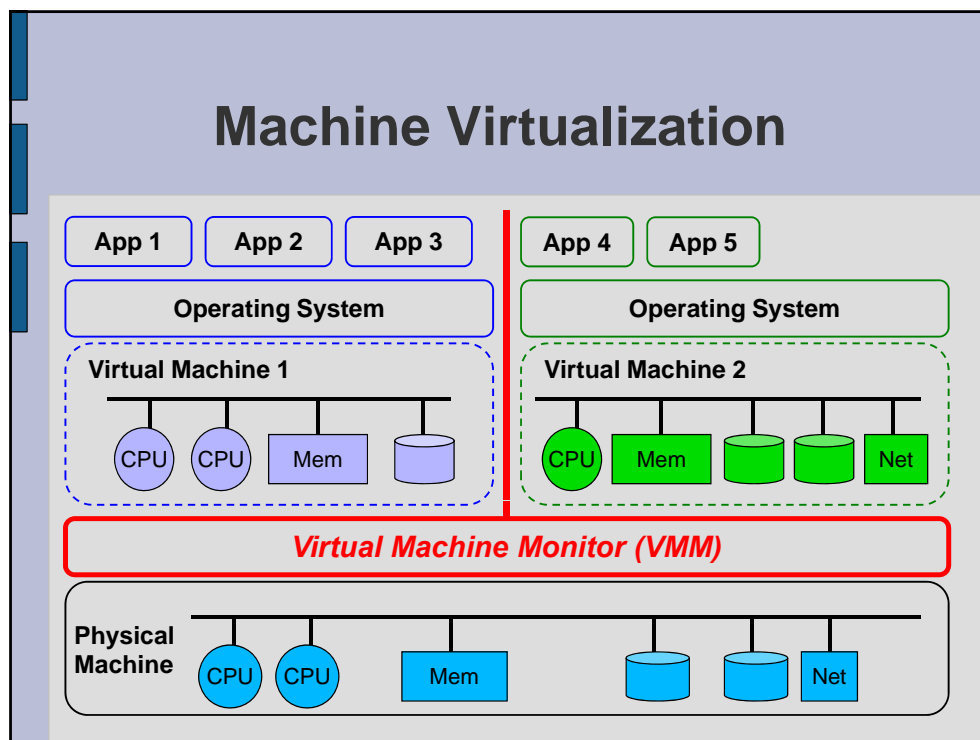
Outline

- Introduction
- Machine Virtualization
 - Overview of machine virtualization
 - Why use virtual machines?
 - Virtual machine technologies
 - Some typical usage scenarios
 - Databases and virtualization
- Storage Virtualization
- Virtualizing the Database Service
- Conclusion

Machine Virtualization

- A **virtual machine** "abstracts" the computing resources of a physical machine into virtual resources
- Introduces a level of **indirection** between virtual resources and physical resources
- End users only see the virtual resources
 - Can install their operating systems and run their applications on the virtual machines
- A **Virtual Machine Monitor** (or **Hypervisor**) is a software layer that implements the mapping from virtual resources to physical resources

Machine Virtualization



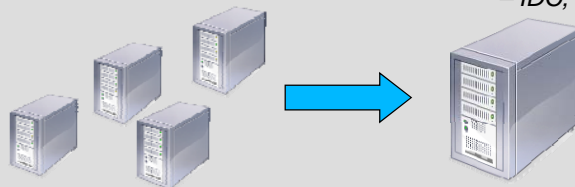
Virtual Machine Monitors

- **Strong isolation** between virtual machines
- **Flexible mapping** between virtual resources and physical resources
 - Can have more virtual resources than the corresponding physical resources
 - Can reallocate physical resources among VMs
- Pause, resume, checkpoint, and migrate virtual machines

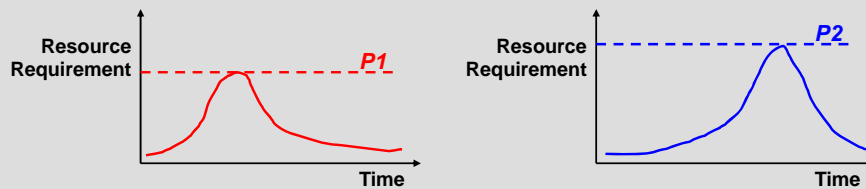
Why Use Virtual Machines?

- Server consolidation
 - Typical setup today: one machine per application (DBMS, web server, mail server, ...)
 - Provisioned for peak load. Usually under-utilized
 - Instead, can run multiple applications on virtual machines that share the same physical machine
 - Save hardware costs and administration/operation costs

"\$140 billion worth of server assets go un-utilized every year"
– IDC, January 2007



Server Consolidation



$$P_{12} < P_1 + P_2$$

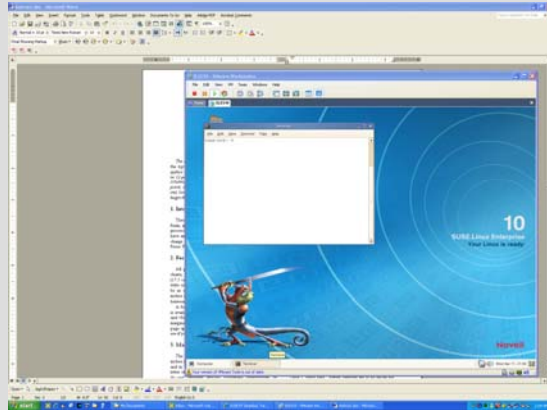
- Consolidate into a single machine with capacity P_{12}
 - Easier to manage
 - Less total capacity than the original two
 - Better utilization than the original two

Why Use Virtual Machines?

- Improved manageability
 - Dynamic provisioning of resources to VMs
 - Migration of VMs for load balancing
 - Migration of VMs to avoid down time during upgrades
- Isolation between VMs
 - Security
 - Privacy
 - Fault tolerance

Why Use Virtual Machines?

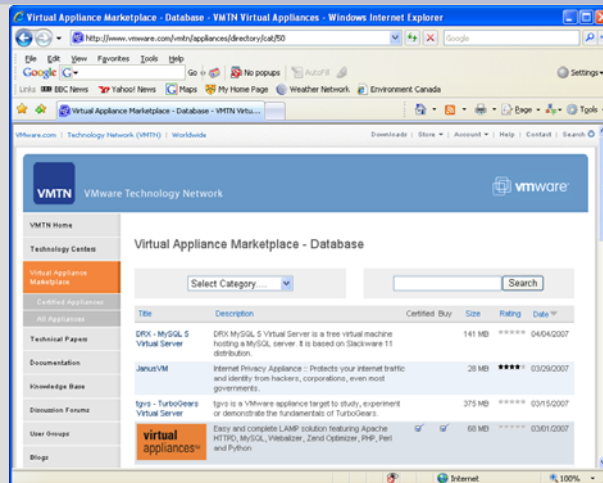
- Application compatibility
 - Different environments for different applications



Why Use Virtual Machines?

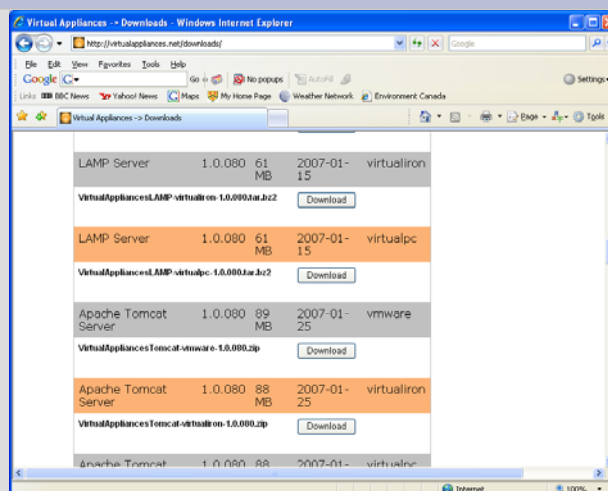
- Software development and testing
 - Multiple environments for development and testing
- Software deployment
 - Preconfigured **virtual appliances**
 - Repositories of virtual appliances on the web

Virtual Appliances



<http://www.vmware.com/vmtn/appliances>

Virtual Appliances



<http://virtualappliances.net/downloads/>

Demonstrations

- 1- Virtualization for **application compatibility**
Linux running on Windows
SUSE Linux Enterprise Server 10
running on Windows XP Professional
using VMware Workstation 5.5.3
- 2- A database **virtual appliance**
Linux running on Windows
Downloading a fully functional ready to run PostgreSQL
server


Why **not** Use Virtualization?

- **Performance penalty**
 - Indirection through VMM adds overhead
- **Hiding details of physical resources**
 - Some applications (e.g., DBMS!) make decisions based on assumptions about the physical resources

To Summarize ...

- Virtualization improves flexibility, manageability, resilience, and interoperability in the computing environment
- The capabilities provided by VMMs can be implemented at other levels, but providing these capabilities at the level of complete virtual machines is simple, useful, and intuitive

Outline

- Introduction
- Machine Virtualization
 - Overview of machine virtualization
 - Why use virtual machines?
 -  – Virtual machine technologies
 - Some typical usage scenarios
 - Databases and virtualization
- Storage Virtualization
- Virtualizing the Database Service
- Conclusion

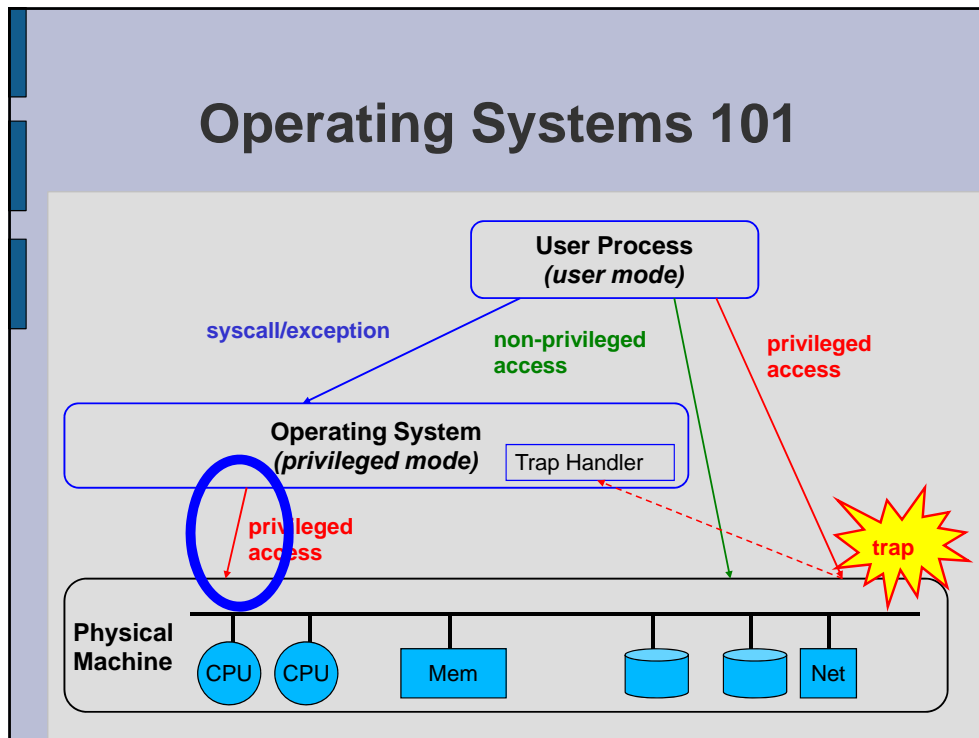
History

- Virtualization has been around since the 1960's [gold74, fidi05]
- Prominent since IBM 370 mainframe series
 - Allowed expensive hardware to be shared by multiple applications running on different operating systems (i.e., server consolidation)
- Virtualization today
 - **Larger scale**
 - **Commodity hardware and operating systems**
 - **Renewed interest in benefits of virtualization**

Operating Systems 101

- Hardware has **privileged** state and instructions
- Operating system (kernel) runs in **privileged mode** and has full access to hardware
- User programs run in **user mode** and need to go through the OS for privileged operations and access (e.g., by making a system call or through an exception handler)
- If a user-mode program tries to perform a privileged operation (e.g., disable interrupts), it **traps** (i.e., causes an exception)

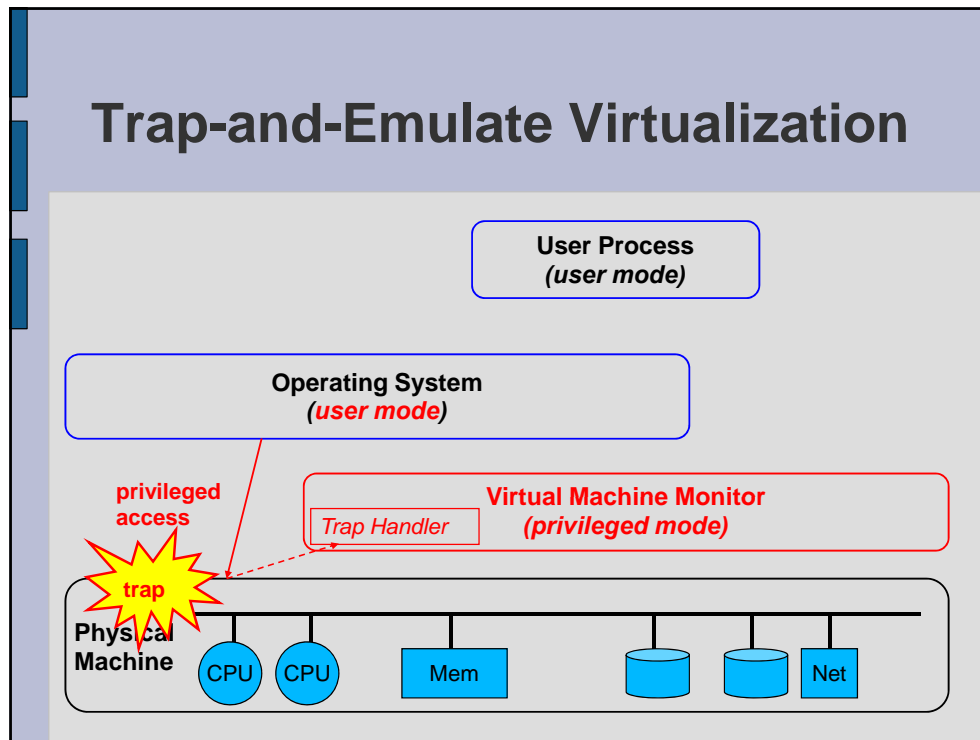
Operating Systems 101



Trap-and-Emulate Virtualization

- Run VMM in privileged mode
- Run OS in user mode
- Privileged operations by the OS will trap
- Trap handler in VMM emulates these operations as if they were run on the virtual machine
- Non-privileged operations can proceed as before with no intervention from the VMM [pogo74]

Trap-and-Emulate Virtualization



Architectural Obstacles

- Some machine architectures are not easy to virtualize
 - Notable example: **x86** [roir00, badr03, adag06]
- Not all privileged operations trap when run in user mode
 - Example: `popf` (pop stack into flags)
 - Privileged mode: change user and system flags
 - User mode: change user flags only, no trap
- Some privileged state is visible in user mode
 - Example: Machine status word
- For an architecture like x86, **trap-and-emulate alone will not work**

Virtualization Approaches

- **Binary rewriting**

- Operating system running in VM is **unmodified**
- VMM scans **Guest OS** memory for problematic instructions and rewrites them
- Example: VMware Workstation [suve01, roga05]

- **Paravirtualization**

- Software interface to VMM is **not identical to hardware**
- Operating systems need to be **ported** to run on VMM
- Simpler VMM and **faster virtual machines** than with trap-and-emulate
- Examples: Denali [whsh02, whco05], Xen [badr03]

Hardware Virtualization for x86

- Intel and AMD have both introduced processor extensions to help virtualization (Intel VT, AMD-V)
- Processor is aware of multiple **virtual machine contexts** (like process control blocks, but for entire operating system)
- New instructions to **start/resume** a VM
- New **privilege level** for VMM
- VMM selects which events should **trap** (`vmexit`)
 - Manipulating interrupt state, interacting with TLB, accessing control registers, ...

Other Architectures

- Other architectures support virtualization and have done so before x86
- IBM POWER
 - IBM System p servers and AIX operating system
- Sun UltraSparc T1

Example Technologies

- *Not a comprehensive list!*
- VMware Workstation
 - Native virtualization of x86
 - Installs as an application on *Host Operating System*
 - Runs unmodified *Guest Operating Systems*
- VMware ESX Server
 - Similar to VMware workstation
 - Installs directly on machine
 - Better performance
 - More control and security



Example Technologies

- Xen

- Paravirtualization of Linux on x86
- VMM could work with other operating systems, but they need to be ported to the Xen VMM



- Virtual Iron

- Based on Xen
- Can combine **multiple physical machines** into **one virtual machine**



- Microsoft Virtual Server

- VMs running unmodified guest operating systems (Windows, Linux) on Windows host



Related Terminology

- **Emulation**

- Simulate complete hardware, allowing unmodified OS for **different CPU** to be run
- Must simulate each instruction so **slow**
- Example: Bochs

- **Operating system virtualization**

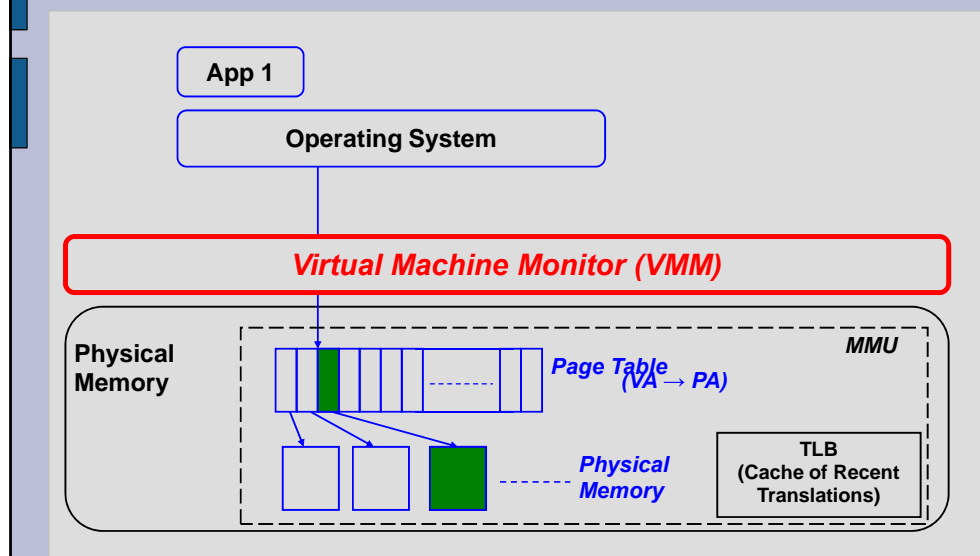
- Isolated virtual servers within a server
- Guest and host OS use **the same kernel**
- Example: Solaris Containers, Virtuozzo

- **What we are discussing is often termed *native virtualization* or *full virtualization*** [smna05]

Example: Virtualizing Memory

- **Virtual address translation**
- Applications (and typically operating system) use virtual addresses
- Operating system and memory management unit (MMU) translate virtual addresses to physical addresses with every memory access

Example: Virtualizing Memory



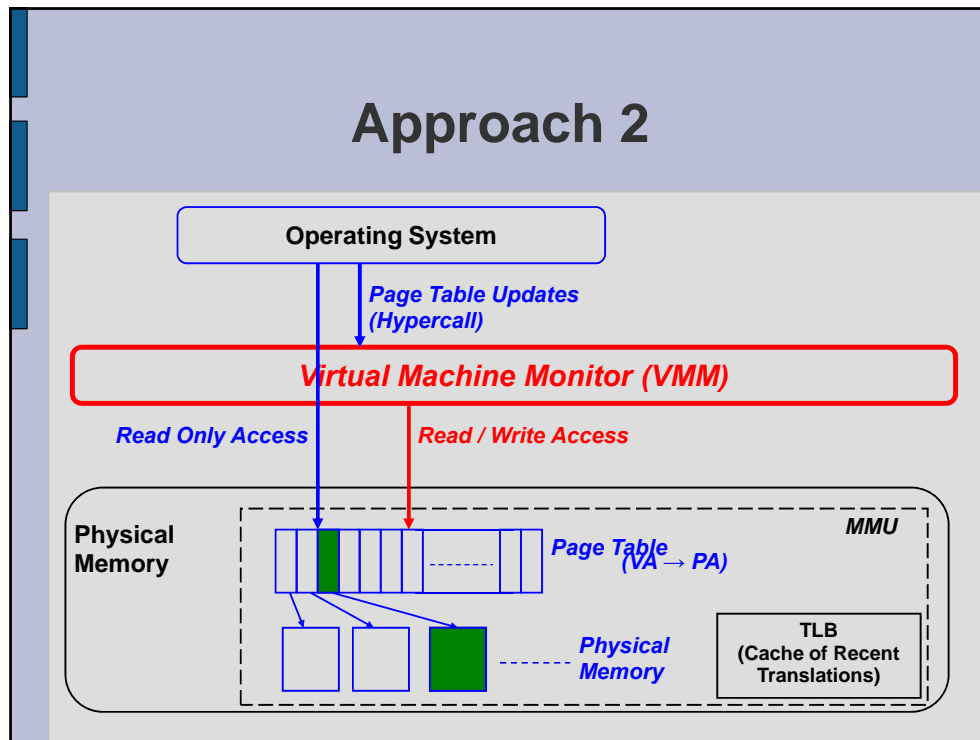
Example: Virtualizing Memory

- Translation Lookaside Buffer (TLB) misses?
- Handled by software
 - Easy: VMM handles TLB miss [engu95]
- Handled by hardware MMU
 - This is the case for x86
 - Difficult: VMM must set up page table so that MMU can use it for address translation
 - **How will Guest OS in VM access this page table?**

Example: Virtualizing Memory

- **How will Guest OS in VM access this page table?**
- Approach 1: Two page tables, one for the OS to see and a **shadow page table** for the MMU to use
 - Used, for example, by VMware
- Approach 2: OS has unrestricted read access to page table and **VMM controls updates** to page table
 - Requires OS modification (i.e., **paravirtualization**)
 - Used, for example, by Xen

Approach 2

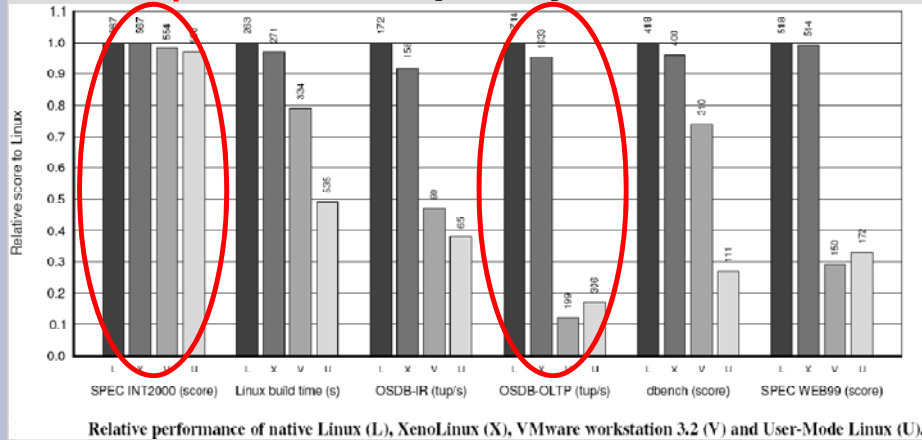


Some Other Issues

- How to give Guest OS illusion of physical memory starting at address 0x00?
- How to deal with reallocating memory, and over-committing memory? [wal02]

Performance Overhead

- Higher use of OS leads to higher overhead
 - *DBMS is a heavy user of OS*
- **Example** results, from [badr03]



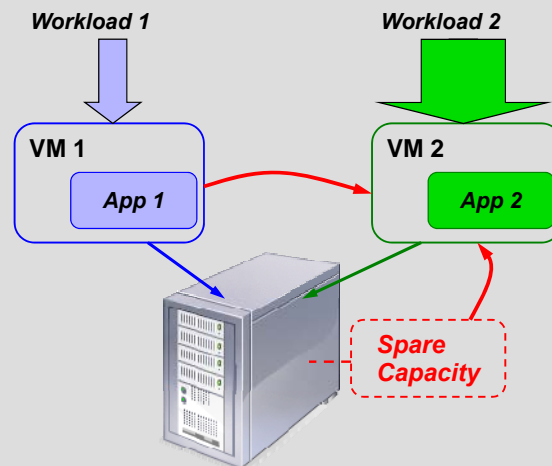
Outline

- Introduction
- Machine Virtualization
 - Overview of machine virtualization
 - Why use virtual machines?
 - Virtual machine technologies
 - ➔ – Some typical usage scenarios
 - Databases and virtualization
- Storage Virtualization
- Virtualizing the Database Service
- Conclusion

Dynamic Resource Provisioning

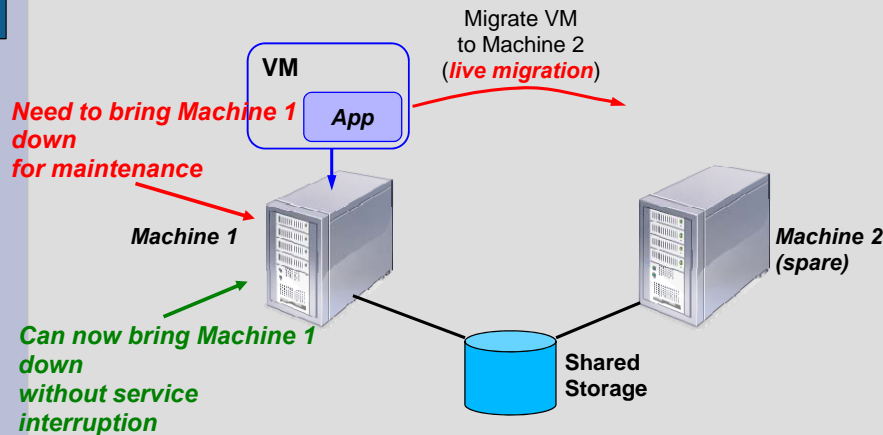
- Several applications providing services to their clients
- Each application runs in its own VM, and the VMs all share the same physical resources
- *As workloads fluctuate, we can dynamically adjust resource allocation levels to get optimal performance*

Dynamic Resource Provisioning



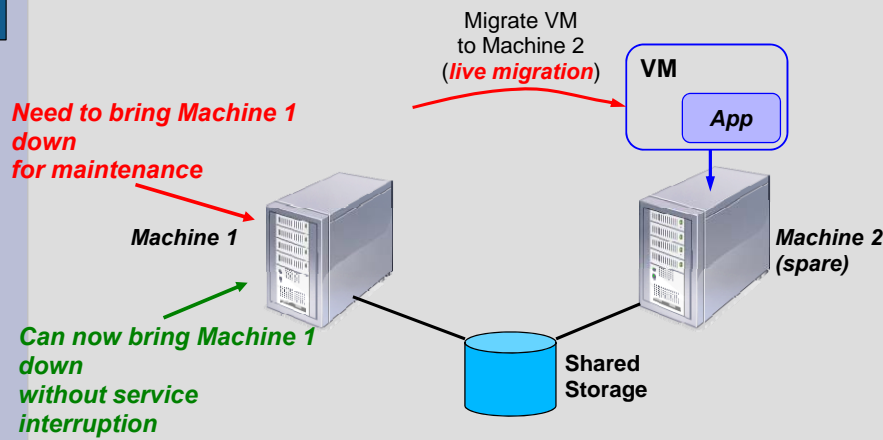
Eliminating Scheduled Down Time

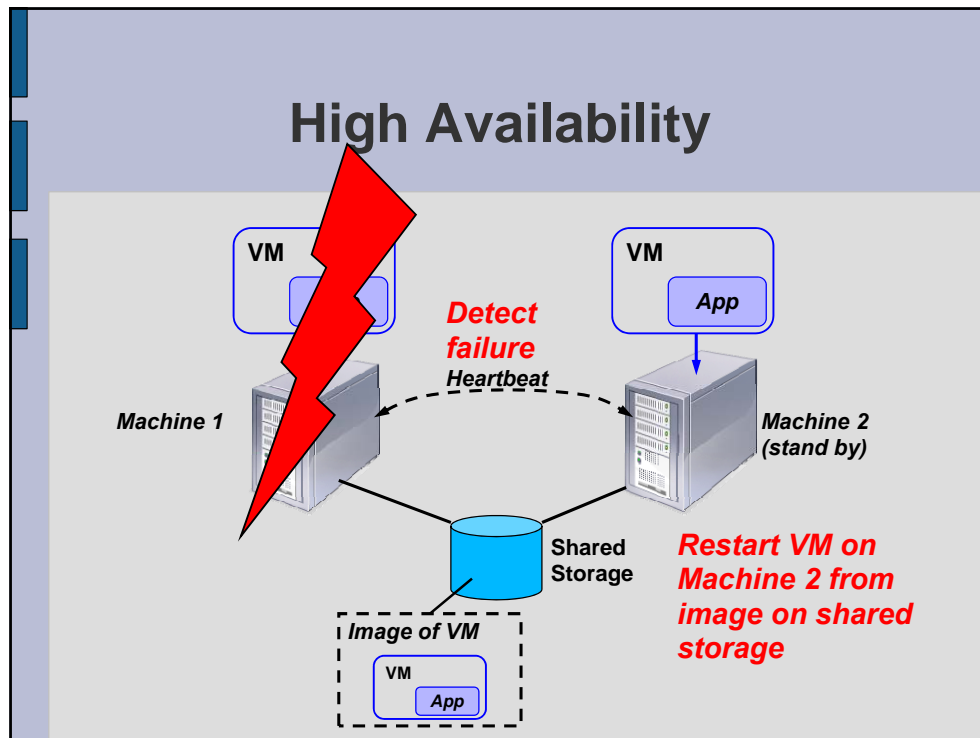
- Use live migration of virtual machines [clfr05]
- Could also be used for **load balancing**



Eliminating Scheduled Down Time

- Use live migration of virtual machines [clfr05]
- Could also be used for **load balancing**





- ## High Availability
- No need for dedicated standby (Can use any machine from "free pool")
 - No need for a-priori setup of failover pair
 - Dynamic resource provisioning can be used to improve overall utilization
 - Schedule workloads on all machines while maintaining a free pool of standby machines
 - Can combine live migration with restart on failure
 - Less need for planned outages
 - Protection against unplanned outages
 - Very high availability

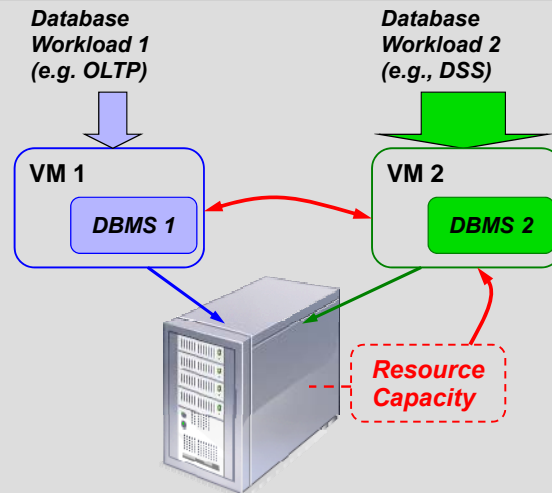
Virtualization and Databases?

- Are database systems just another application running in the virtualized environment?
- Is there anything that needs to be done for database systems to run well in virtualized environments?
- ***Virtualization poses several interesting research questions for database systems***
 - Revisit the previous three scenarios in the context of database systems

Database System Agility

- Environment in which database system is running can change rapidly and dramatically
 - CPU power
 - Number of CPUs?
 - Available memory
 - Available I/O and network bandwidth
- Database system needs to be able to effectively adapt to such changes while it is running
- This is also needed for non-virtualized environments
- However, with virtualization, the potential for changes in the environment is higher

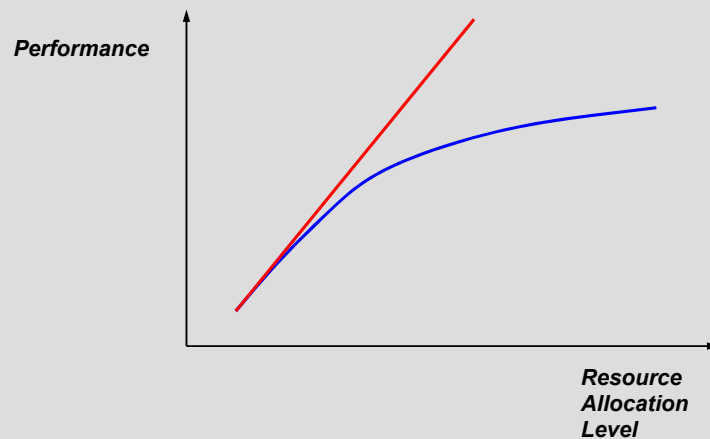
Resource Provisioning



Resource Provisioning

- **What level of resources to give to each DBMS?**
 - Configuring VM parameters
- **How to tune the DBMS for a given level of resources?**
 - Configuring the DBMS parameters
- A resource allocation problem
- Need a **model** of how resource allocation affects database performance
- Need optimization or control algorithms to decide on the optimal resource allocation [pazh07]

Performance Model

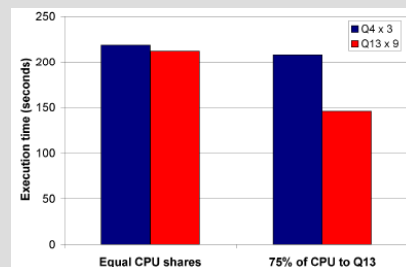
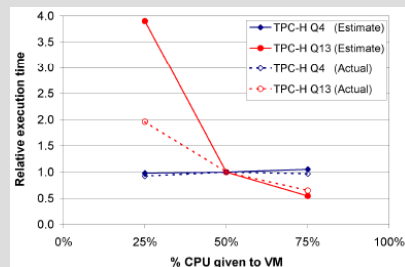


Static Version

- Static version of the resource provisioning problem: Workloads and resources known in advance
- **Database appliance configuration**
 - Necessary if we want to use virtualization as an effective tool for software deployment
- A **more constrained** version of the **capacity planning** problem
 - Instead of "How big does my machine need to be?" we ask "How much **of the available resources** to give to each machine?"

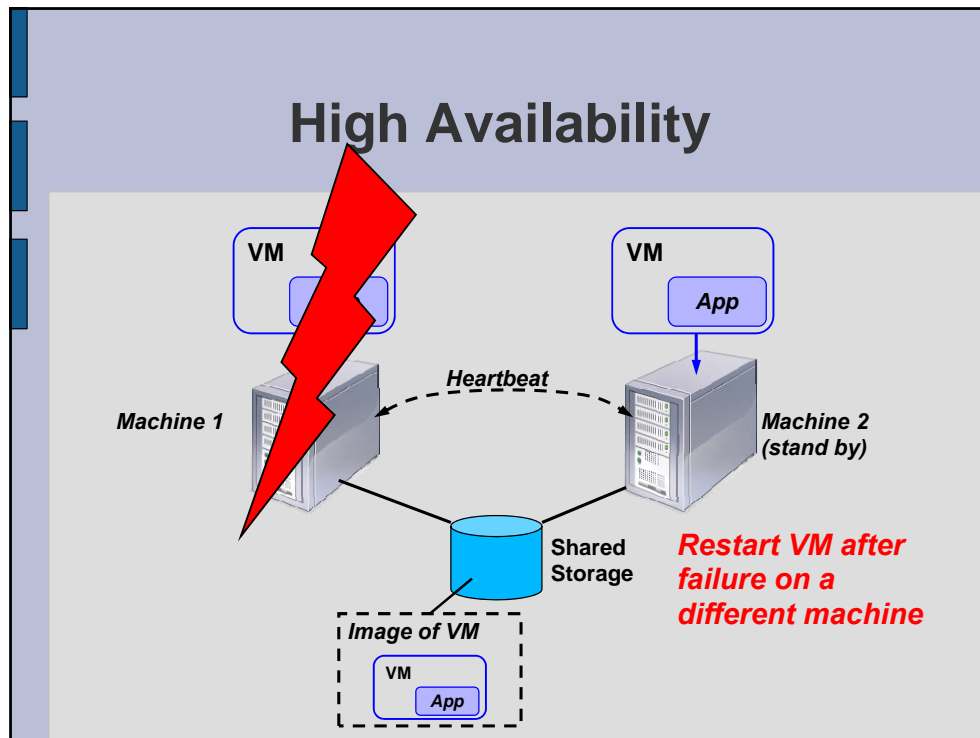
Example Approach

- Use **query optimizer** as the cost model [soab07]
 - Calibrate it to reflect virtual machine resource levels
- Example (from [soab07]): Xen VMs running TPC-H queries on PostgreSQL and sharing the same physical machine
 - What is the best CPU allocation?



Resource Provisioning Issues

- **Accurate modeling of performance**
 - Concurrent queries within a VM
 - Time varying workloads
 - Non-database workloads
- **Performance objectives**
 - Service Level Agreements
 - Deadlines / time outs / target response times
 - Resource utilization levels
- **Determining the optimal resource allocation**
 - Combinatorial search
 - Automatic control
- **Database system agility**




High Availability

- Problem: Need to protect **large amounts of state**
 - Database
 - DBMS processes
- Also a problem for live migration of DBMS
- Can we restart from the original VM image?
- Can we do better by having more recent images?
- Can we speed up database recovery after restart?
- Can we avoid losing connections and buffer pool?
- Can we leverage work on persistent DBMS sessions [lowe98, balo00, balo04]?
- Can we do this without shared storage?

Special APIs

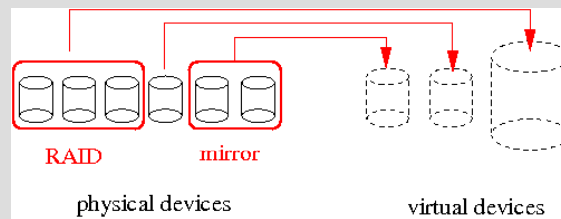
- Can we benefit from special APIs between DBMS and VMM?
 - These APIs may be useful for other applications as well
- **Hints** between DBMS and VMM
 - Performance objectives
 - Resource allocation constraints
 - Consistency points
 - ...
- Special APIs for dealing with large amounts of state
 - Checkpointing

Outline

- Introduction
- Machine Virtualization
-  • **Storage Virtualization**
 - What is storage virtualization?
 - Why use it?
 - Implementations of storage virtualization
 - Challenges and opportunities for DBMS
- Virtualizing the Database Service
- Conclusion

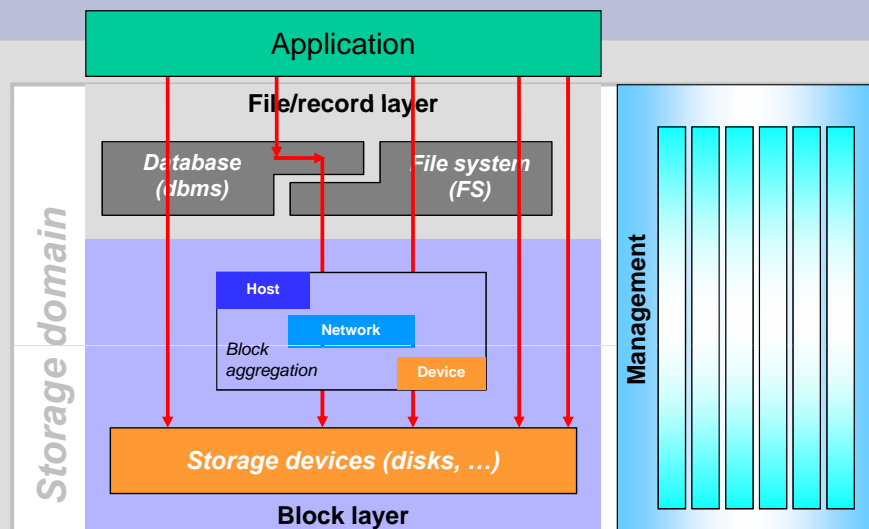
What is Storage Virtualization?

- storage virtualization is a layer of indirection that allows the definition of virtual storage devices



- virtualization isolates storage clients from the physical reality of the storage system

SNIA Shared Storage Model



Copyright © 2000-2003, Storage Networking Industry Association

Basic Capabilities of Virtual Storage

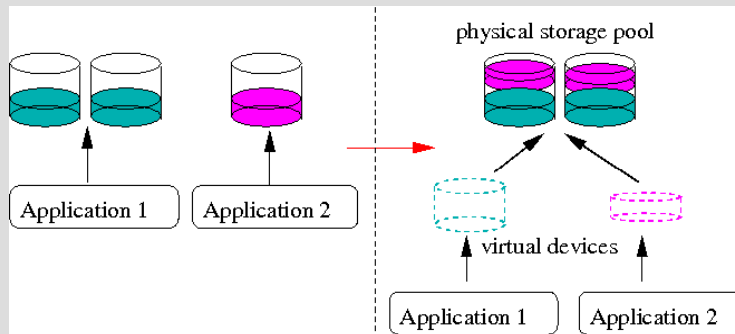
- create, destroy virtual devices using available *pool* of physical storage
- grow, shrink virtual devices
- control properties of virtual devices
 - size
 - performance
 - reliability
- dynamic provisioning of physical storage

Additional Capabilities of Virtual Storage

- versioning, snapshots, point-in-time copies
- local and remote mirroring
- migration of virtual devices
 - supports provisioning, hierarchical storage management
- auto-administration
 - policy-based management
- storage QoS and performance isolation
 - active research area: [kaka05, utyi05, hape04, wech04, goja03, lume03, brbr99]

Why Virtualize Storage?

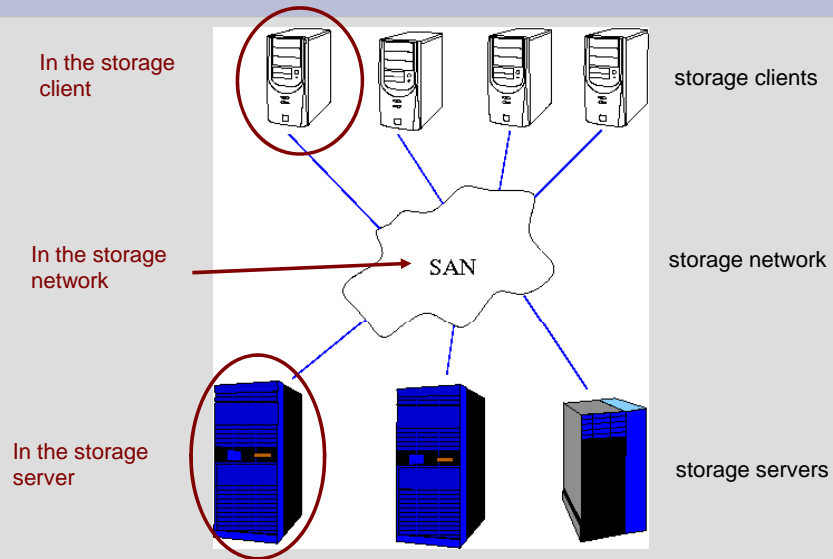
- improve storage utilization
- reduce storage costs



Why Virtualize Storage?

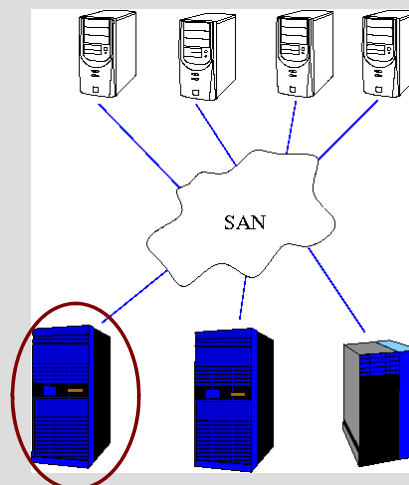
- minimize/avoid downtime
 - simplify maintenance tasks
 - transparent redundancy
- improve performance
 - distribute and balance storage loads
 - dynamic storage provisioning
 - control placement
- reduce cost of storage administration
 - single point of administrative control
 - simplified operations
 - automation, e.g., policy-based management

How to Virtualize Storage



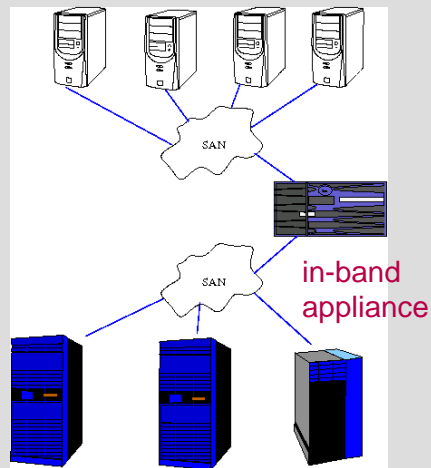
Virtualization in Storage Servers

- virtual devices limited to a single server
- storage server coordination, e.g., mirroring



Virtualization Appliances

- integrate heterogeneous servers
- centralized administration
- potential performance bottleneck

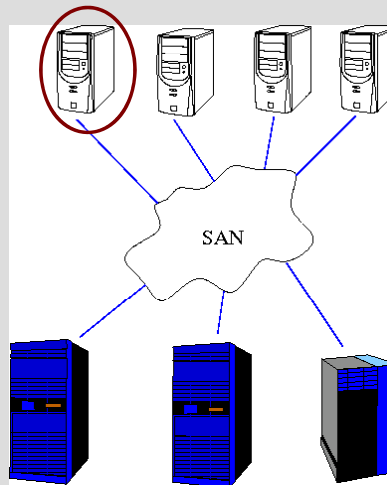


Example: HP StorageWorks SVS200

- in-band appliance
- pooling of heterogeneous storage servers
- transparent data migration
- local and remote mirroring
 - 1-safe and 2-safe mirroring
 - split mirrors

Virtualization in Storage Clients

- via *logical volume management* (LVM) in the storage client
- e.g., Linux LVM2
 - create, destroy, resize, snapshot, migrate logical volumes



Limitations of Logical Volume Management

- administrative scalability
 - separate LVM in each storage client
 - no single point of administrative control
- heterogeneity
 - different LVMs in different clients
- inter-client allocation
 - additional volume management is required to reallocate resources among storage clients

The DBMS Perspective: So What?

- Virtual storage devices are **dynamic**
 - affects DBMS storage management, e.g., resizable tablespaces
- Virtual storage devices are **opaque**
 - affects DBMS configuration and tuning
- Virtual storage devices are based on **shared physical resources**
 - separate administration, distinct goals
- Virtual storage devices **more capable** than physical devices
 - CPU/memory, functionality

DBMS Configuration and Tuning

- characteristics of physical storage hidden from the DBMS (and DBA)
- how to:
 - layout DB objects?
 - set DBMS parameters?
 - page/extent sizes
 - prefetching
 - costs

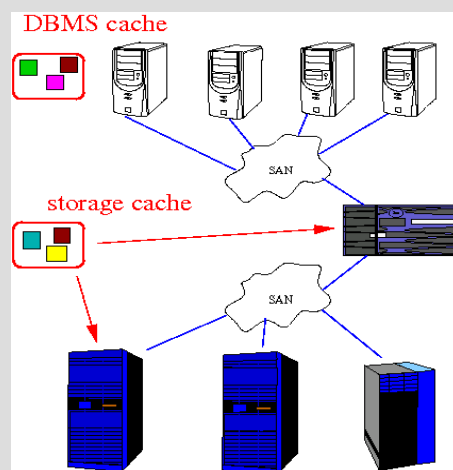
```
CREATE TABLESPACE ts
MANAGED BY DATABASE
USING (
  DEVICE '/dev/rlv1' 10000,
  DEVICE '/dev/rlv2' 10000)
OVERHEAD 6.0
TRANSFERRATE 0.05
PAGESIZE 8192
```

Exploiting Storage Capabilities

- storage snapshots and versioning
 - backup and recovery
 - concurrent applications
- storage replication and mirroring
 - dynamic DBMS provisioning
- dynamic storage resource allocation
 - accommodate workload fluctuations
- CPU and memory in the storage system
 - caching
 - offload DBMS functions

Multi-Tier Caching

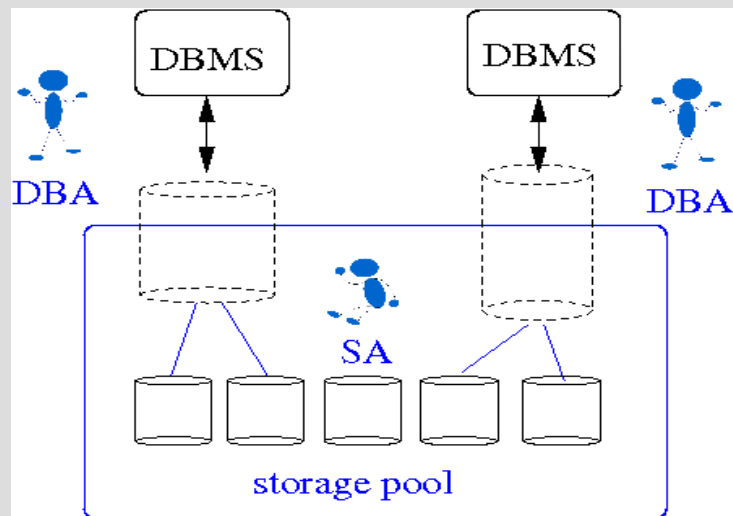
- exploit memory in the storage system
- DBMS can improve 2nd tier cache performance
- example: hinting [liab05]



DBMS and Storage Administration

- Textbook:
 - DBA understands and controls dedicated physical devices
- Reality:
 - storage is virtual
 - virtual storage is separately administered
 - DBA and storage administrator (SA) must coordinate

DBAs and SAs

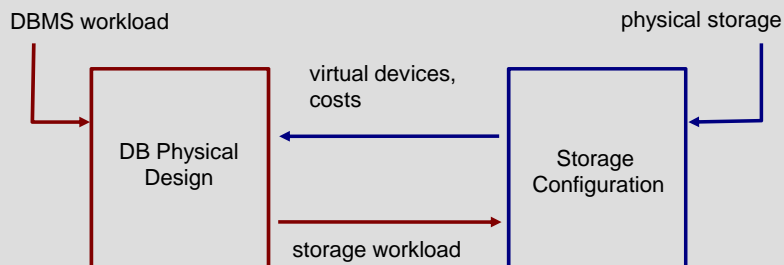


Tasks of DBAs and SAs

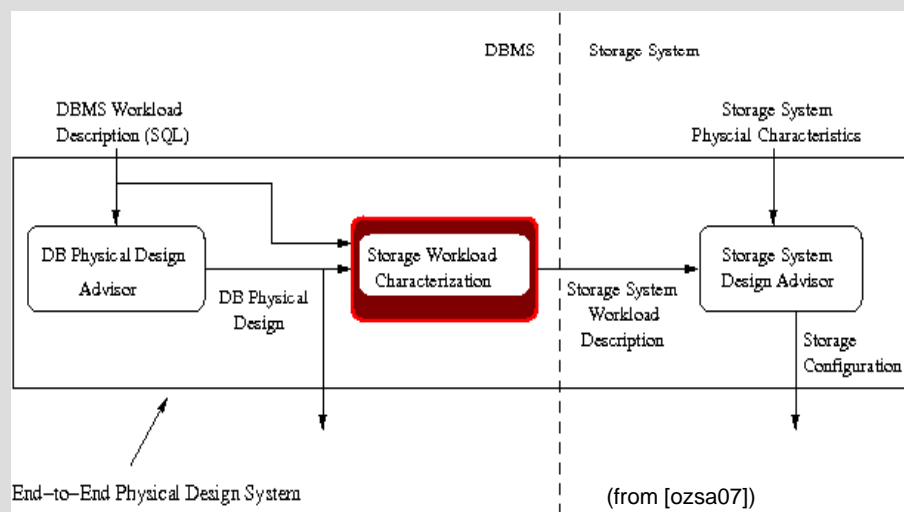
- database administration
 - DB physical design [brch05, coba05, vazu00, agch00]
 - layout [agch03]
- storage administration:
 - define and configure virtual devices
 - storage allocation, capacity planning
 - tools [anho02, waos02, anka01, albo01, dech03]

DB+Storage Co-Design

- DB Physical Design
 - index selection, layout
- Storage Configuration
 - define, layout, configure virtual devices



Storage Workload Estimation

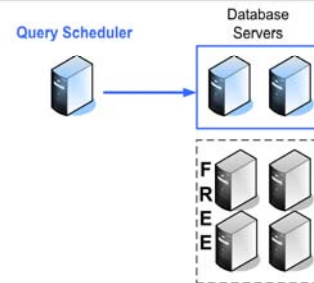


Outline

- Introduction
- Machine Virtualization
- Storage Virtualization
- ➔ • Virtualizing the Database Service
 - What is Database Service virtualization?
 - Why use it?
 - Technical challenges and Case studies
 - Opportunities for DBMS researchers
- Conclusion

What is DB Service Virtualization ?

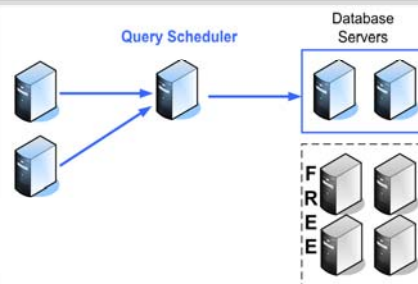
- A scheduling (indirection) tier which makes the DB service look like a single, **scalable and available** DBMS to clients
- Enables transparent changes to the mapping of application resource allocations to physical servers



Scheduling tier virtualizes database cluster

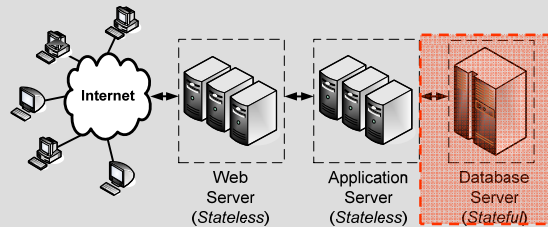
Why Virtualize the DB Service ?

- Transparently adapt to
 - changing load, by dynamic server provisioning
 - failures, by service migration
- Meet application Service Level Agreements
 - E.g., latency, throughput requirements



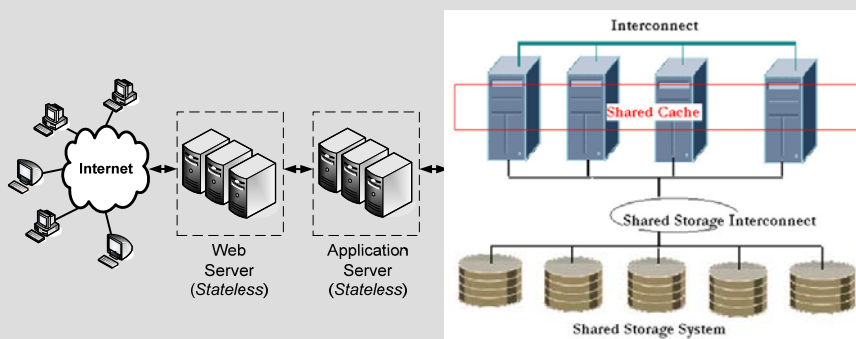
Transparent scaling and fail-over for DB backend

Useful in Data Centers



- Virtualization of front-end already present
 - through dynamic server provisioning e.g., IBM's Tivoli, WebSphereXD, [Benn05], [Urga05], [Kar06]
- Database tier: typically, single over-provisioned node or Oracle RAC cluster.

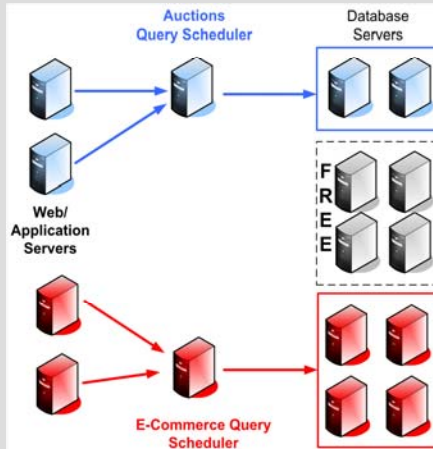
Server Farm with Oracle RAC



- Network attached shared storage solution, allows CPU provisioning in the DB back-end [Lah01]

Benefit of Service Virtualization based on Commodity Servers

- Resource consolidation in complex server farms
 - Data centers running multiple applications
 - Resource multiplexing if different peak times
- Reduce costs
 - Management and cooling

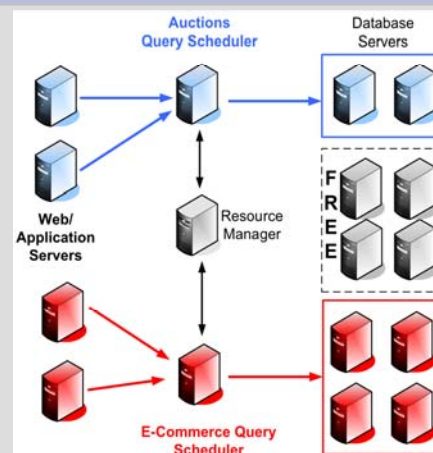


Virtualization allows resource consolidation for commodity servers

Virtualized DB Tier Architecture

Technical challenges

- State management ←
- Replica allocation
 - Decide how many instances of each application should run
- Replica mapping
 - Decide on which node each instance should run

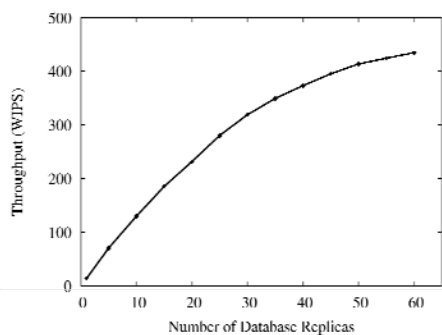


Resource manager controls allocations and mapping

State Management Approaches

- Use full database replication
- Asynchronous replication with consistency guarantees
[Platt04], [Lin05], [Amza05], [Da06]
- Read-one write-all workload scheduling
 - within each application's allocation

Why Database Replication ?



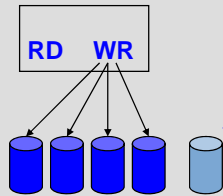
Scaling for E-Commerce (TPC-W)
[Amza03]

Shown to scale well

- [Platt04] MW'04
- [Lin05] SIGMOD'05
- [Amza05] ICDE'05
- [Da06] VLDB'06

Unified approach to load peaks and fault handling

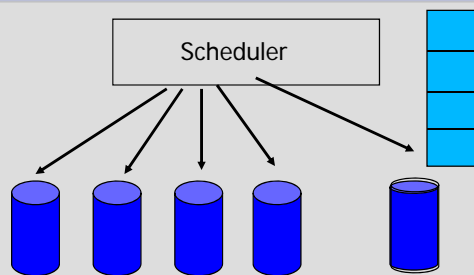
State Management Challenges



Adaptation steps:

1. Updating replica [Das05], [Kemme01], [So06]
2. Load balancing and buffer pool warm-up

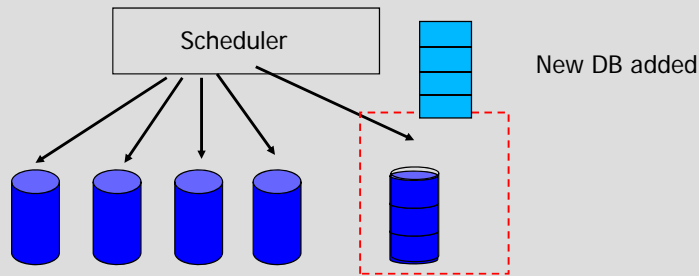
Updating replica



Update log replay on new replica

When do we start sending new updates to replica ?

Updating Replica



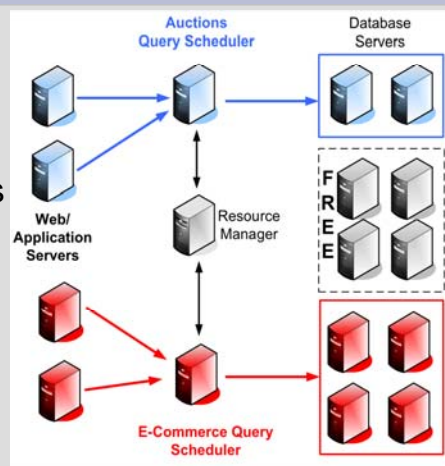
- Performed in phases [Kemme01], [So06], [SA06]
- In each phase, a portion of the log is applied on new replica
- During last phase, new update queries can be queued as well
- After replica is fully up to date, we apply the queued updates

Other optimizations: throttled data migration [Das05]

Virtualized DB Tier Architecture

Technical challenges

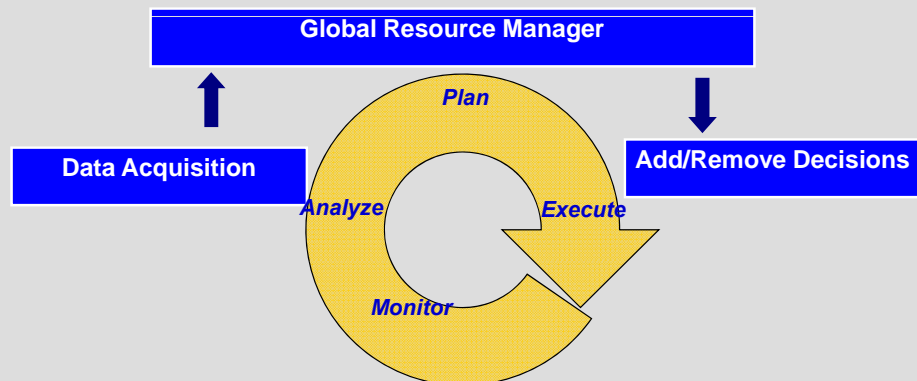
- State management
- Replica allocation ←
- Decide how many replicas of each application should run
- Replica mapping
- Decide on which node each instance should run



Replica Allocation Approaches

- Reactive database allocation
 - Based on feedback loop [So06,SA06]
- Proactive database allocation
 - Based on system modeling
 - Analytical Models [Urga05], [Benn05], [Wood06]
 - Utility-based approaches [Wal04], [Tes05]
 - Machine learning approaches [Tes06], [Chen06], [Ghan07]

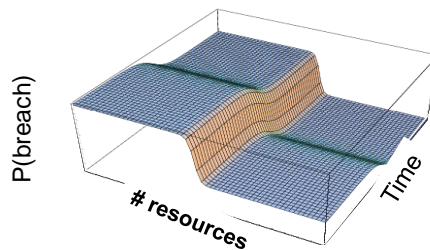
Reactive Replica Allocation



Incrementally add/remove (one) replica based on periodic sampling

Proactive Replica Allocation

- Build model from system states (on/off-line)
- Predict load, determine configuration to accommodate predicted load

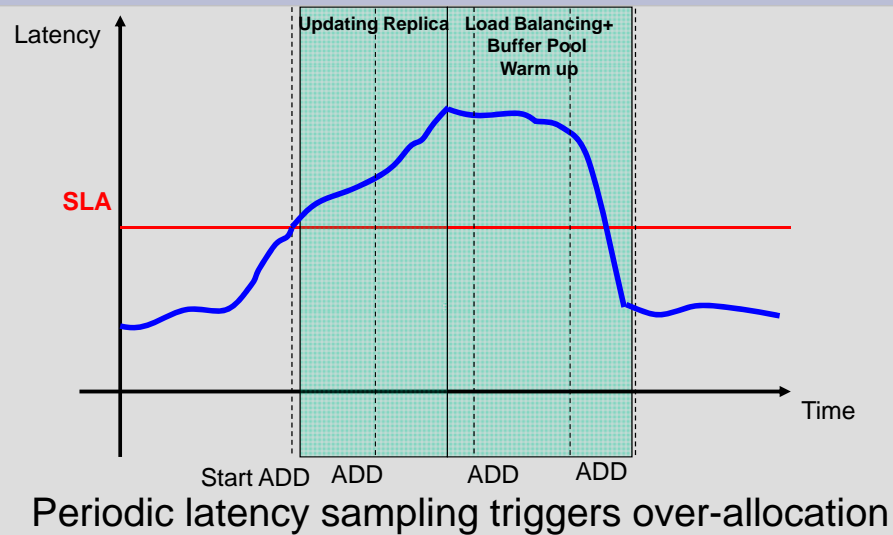


Add or remove several database replicas at once

Replica Allocation Challenges

- Potentially high adaptation delay
- Reactive approaches may over-allocate, hence oscillate
 - due to feedback delay
- Proactive approaches may be imprecise
 - due to need for long term prediction
 - also subject to allocation oscillations

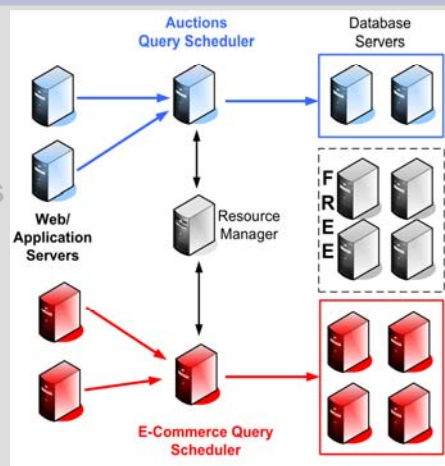
Replica Allocation Oscillation



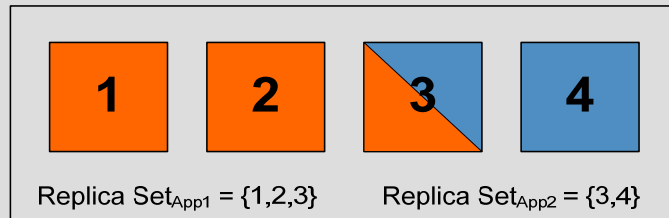
Virtualized DB Tier Architecture

Technical challenges

- State management
- Replica allocation
 - Decide how many replicas of each application should run
- Replica mapping ←
 - Decide on which node each instance should run

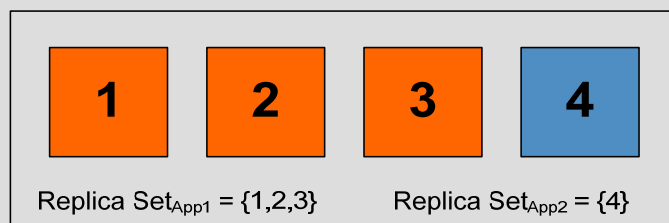


Replica Mapping



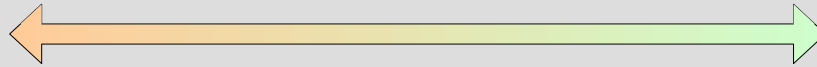
- Mapping allocations to machines
- Two options
 - overlapped replica sets
 - disjoint replica sets

Replica Mapping



- Two options
 - overlapped replica sets
 - disjoint replica sets

Replica Mapping Challenges

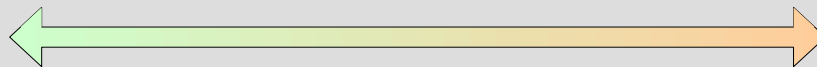


Disjoint
SLOW
Adaptation

Full Overlap
NO
Adaptation

- Disjoint: Adding a replica can take a long time
 - Bring replica up-to-date
 - Warm-up memory

Replica Mapping Challenges

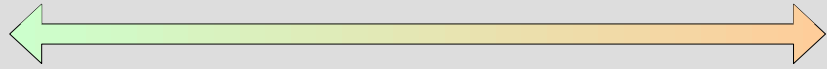


Disjoint
NO
Interference

Full Overlap
HIGH
Interference

- However, overlapping applications compete for resources causing interference for
 - CPU, memory, L1/L2 cache, storage cache hierarchy

Replica Mapping Challenges



Disjoint
NO
Interference

Full Overlap
HIGH
Interference

Tradeoff between adaptation delay and interference

Option: overlap, but control the interference by
informed query co-scheduling and quota enforcement
e.g., using VMM

How to Address Challenges ?

- Prevent Oscillations
 - Take adaptation instability into account
- Reduce adaptation delay
 - Proactively allocate several replicas at once
 - Update additional replicas
 - Pre-warm buffer pool
- Avoid Interference
 - Co-schedule queries and enforce quotas

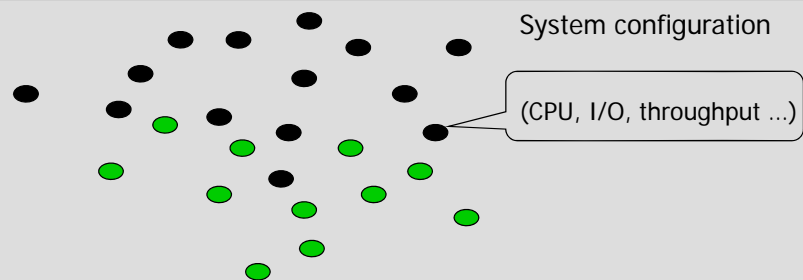
How to Prevent Oscillations by Instability Detection

- Learn normal range of metrics and their correlations
 - e.g., load imbalance range, query throughput vs. active connections, or throughput variation vs. latency variation
- Online detection
 - Suppress actions during instability
 - [So06], [Chen06], [Ghan06]

How to Reduce Adaptation Delay by Proactive Allocation

- Add database replicas to application
 - Proactive with off-line learning [Chen06]
 - Proactive with on-line learning [Ghan06]

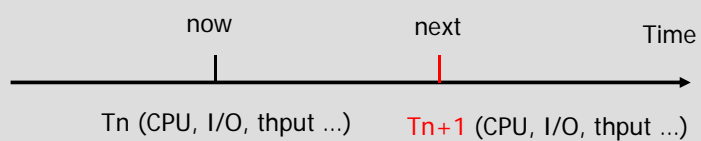
Case Study 1: Proactive with Off-line Training



- Measure system and application metrics
- Under stable system configurations (# of dbs)

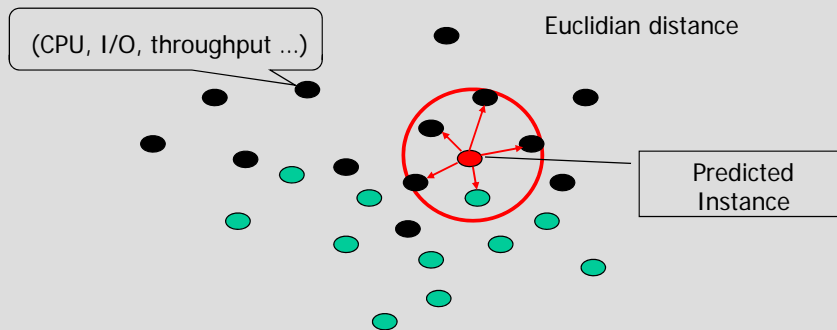
[Chen06]

Online Prediction



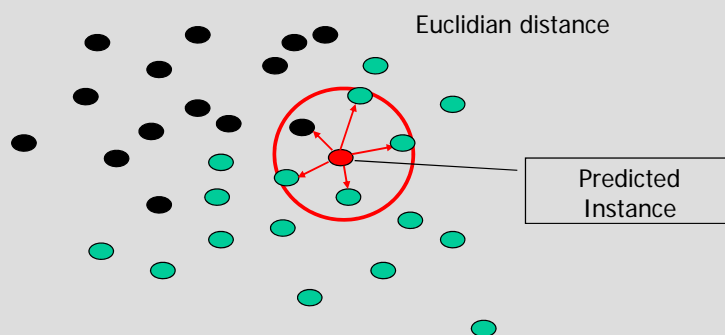
- Predict metrics in the next interval
- Match predicted instance to offline training data

On-line Classification



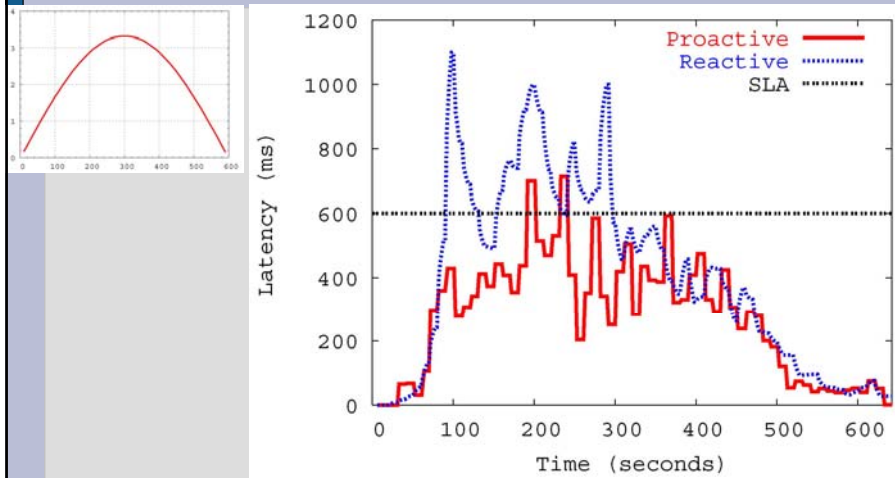
SLA will be violated in the next interval

Determine New Configuration



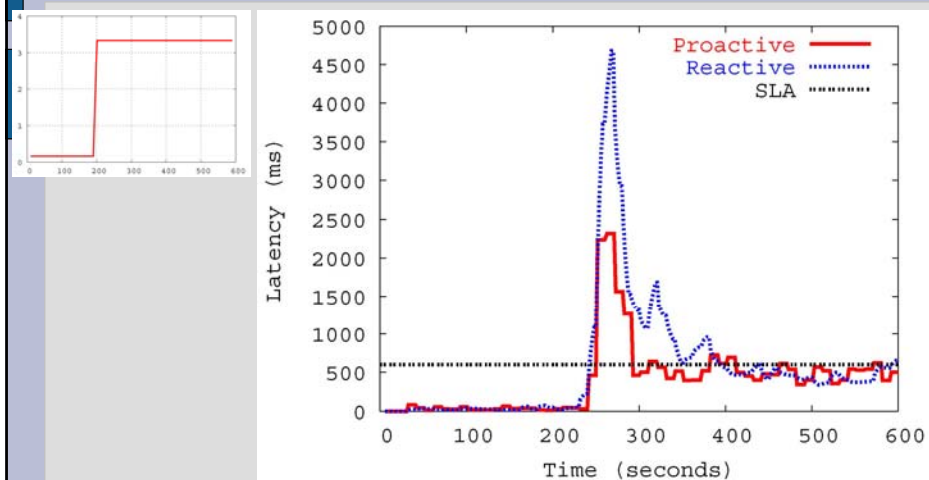
Search configuration (# of dbs) for which SLA is met
Add databases to allocation

Proactive vs. Reactive Allocation



What if load intensity is unpredictable ?

Unpredictable Load Spike



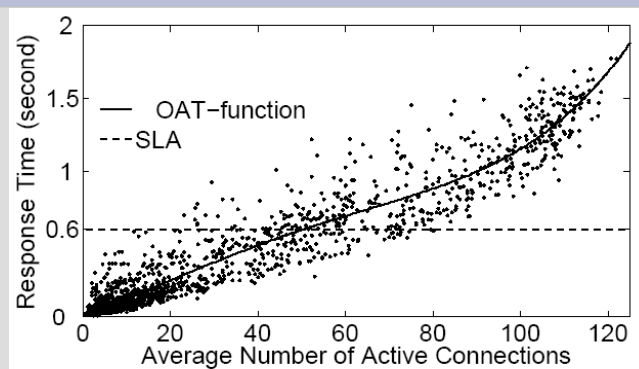
What if workload mix changes ?

Case Study 2: Proactive with On-line Training

- Collect sample set of states on-line
- Learn correlation function between metrics
$$\text{Latency} = f(m1, m2, \dots \#dbs) \quad [\text{Ghan07}]$$
- Solve constrained optimization problem to get #dbs
- Allocate database replicas proactively

Adapt to workload mix change by dynamically evolving sample set

Case Study 2: Proactive with On-line Training



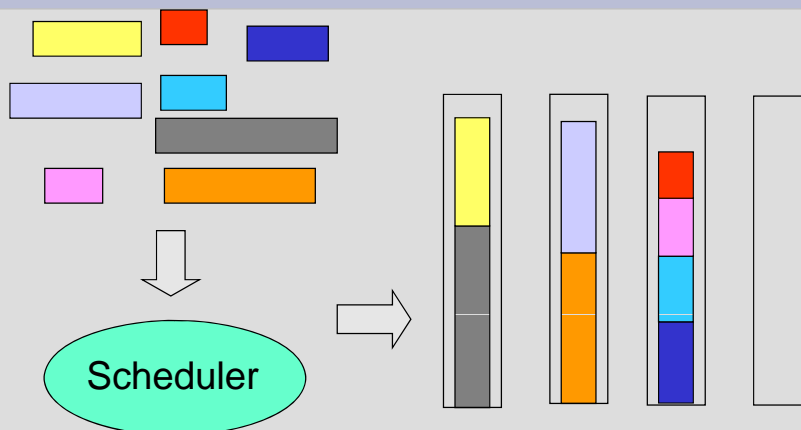
Example correlation function [Ghan07]

How to Avoid Interference Using Informed Co-scheduling

Fine grained resource allocation

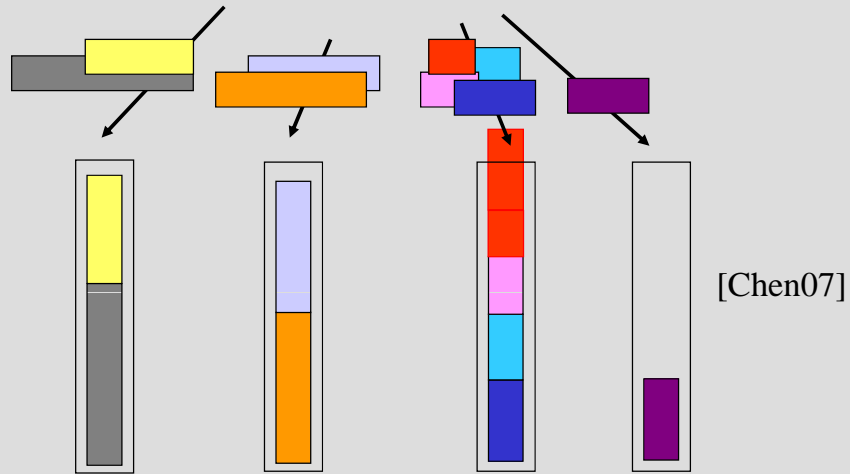
- Use cost models per query to estimate resource consumption (CPU, memory)
- Bin-pack and selectively schedule queries

How to Avoid Interference when Co-scheduling



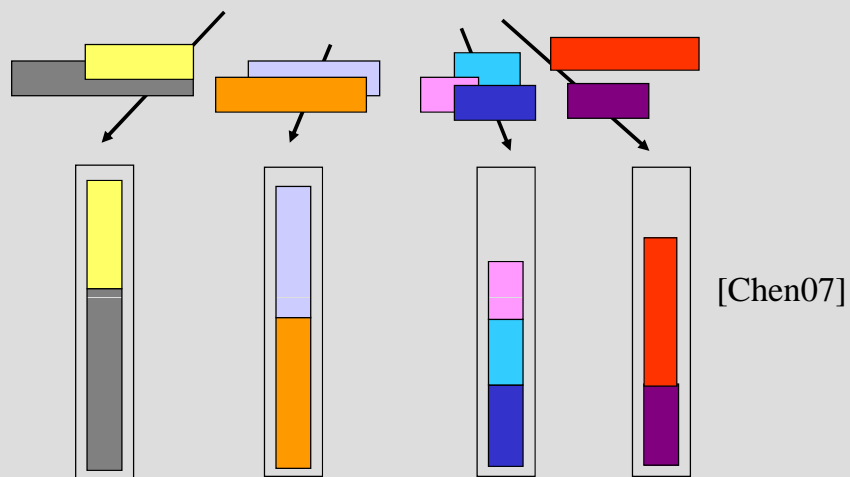
Per-query cost models for CPU, memory, I/O
[Elmi07] [Chen07][Sor07] [Qin07]

Fine-grained Query Scheduling



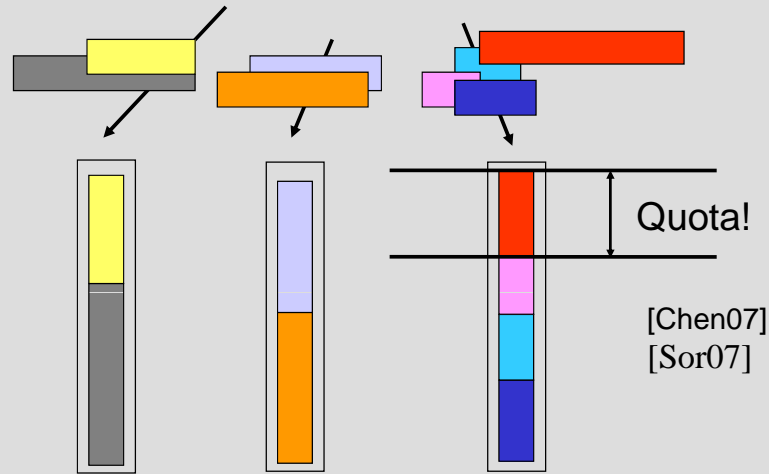
Query Scheduling based on Resource Usage Model

Fine-grained Query Scheduling



Selective Rescheduling

Quota Enforcement



Quota Enforcement for Memory and CPU


Conclusions: DB Service Virtualization Benefits

- Automatically meet SLA for each application
 - Performance
 - Data availability
- Good resource usage
 - Resource multiplexing for applications and tiers
 - Reduced operational costs

Opportunities for DB Researchers

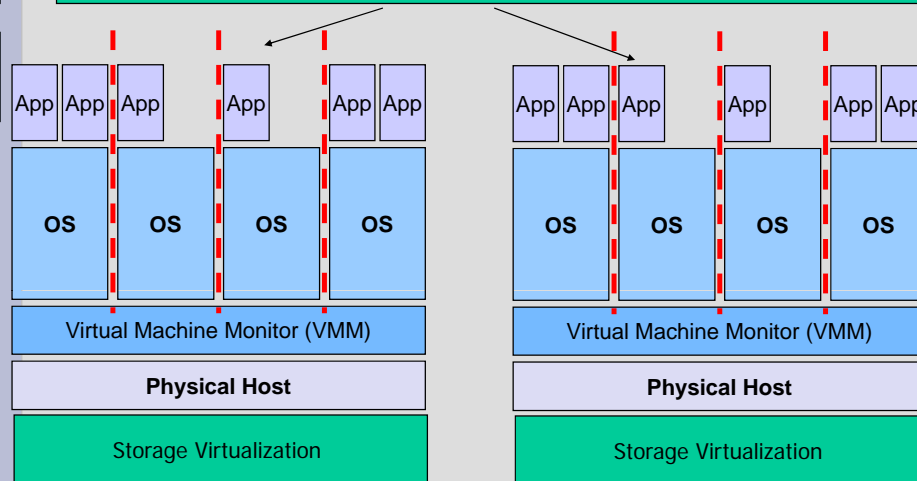
- High impact practical problem
- Classic DB techniques applicable to problem
- Data mining techniques
 - mine stream of monitored metrics for correlations
- Co-scheduling for interference avoidance
 - e.g., dynamic buffer pool management
[Brown93],[Brown96],[Mart06],[Storm06],[Bowm07]
 - How do we define quotas based on SLA ?

Outline

- Introduction
- Machine Virtualization
- Storage Virtualization
- Virtualizing the Database Service
-  • Conclusion

Trend: Virtualization at Many Levels

Server, Storage, and Service Virtualization



Database Research Challenges

- Database-aware virtual resource configuration
- Modeling database performance in virtualized environments
- Using virtualization to improve availability and fault tolerance
- Database administration in virtualized environment
- Exploiting computational and memory resources of virtualized storage
- Exploiting other capabilities of virtualized storage (snapshots, mirroring, replication, ...)
- Special APIs between DBMS and virtualized environment
- Dealing with variations in the type and intensity of the database workload

Conclusion

- Virtualization
 - Provides powerful mechanisms for solving many current problems in the computing infrastructure
 - Increasingly being supported and adopted by a wide range of organizations
- Virtualization and database systems
 - Significantly changes the operating environment
 - But at the same time can be very useful
- ***Many opportunities for database researchers***

References

- [adag06] Keith Adams and Ole Agesen. A Comparison of Software and Hardware Techniques for x86 Virtualization. International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), October 2006.
- [agch00] Sanjay Agrawal, Surajit Chaudhuri, and Vivek R. Narasayya. Automated selection of materialized views and indexes in SQL databases. In Proc. International Conference on Very Large Data Bases, pages 496-505, 2000.
- [agch03] Sanjay Agrawal, Surajit Chaudhuri, Abhinandan Das, and Vivek Narasayya. Automating layout of relational databases. In International Conference on Data Engineering (ICDE '03), pages 607-618, 2003.
- [albo01] Guillermo A. Alvarez, Elizabeth Borowsky, Susie Go, Theodore H. Romer, Ralph Becker-Szendy, Richard Golding, Arif Merchant, Mirjana Spasojevic, Alistair Veitch, and John Wilkes. Minerva: An automated resource provisioning tool for large-scale storage systems. ACM Transactions on Computer Systems, 19(4):483-518, 2001.
- [Amza03] C. Amza, A. L. Cox and W. Zwaenepoel. Conict-Aware Scheduling for Dynamic Content Applications. In USENIX Symp. on Internet Tech. and Sys. 2003.
- [Amza05] C. Amza, A. Cox, W. Zwaenepoel. A Comparative Evaluation of Transparent Scaling Techniques for Dynamic Content Servers. In ICDE '05.

References

- **[anho02]** Eric Anderson, Michael Hobbs, Kimberly Keeton, Susan Spence, Mustafa Uysal, and Alistair Veitch. Hippodrome: running circles around storage administration. In Conference on File and Storage Technology (FAST'02), pages 175-188, January 2002.
- **[anka01]** E. Anderson, M. Kallahalla, S. Spence, R. Swaminathan, and Q. Wang. Ergastulum: quickly finding near-optimal storage system designs. Technical Report HPL-SSP-2001-5, HP Laboratories, July 2001.
- **[badr03]** Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. Xen and the Art of Virtualization. ACM Symposium on Operating Systems Principles (SOSP), pp. 164-177, October 2003.
- **[balo00]** Roger S. Barga, David B. Lomet, Thomas Baby, Sanjay Agrawal: Persistent Client-Server Database Sessions. International Conference on Extending Database Technology (EDBT), pp. 462-477, March 2000.
- **[balo04]** Roger S. Barga, David B. Lomet, German Shegalov, Gerhard Weikum: Recovery guarantees for Internet applications. ACM Transactions on Internet Technology 4(3): 289-328, August 2004.
- **[Benn05]** M. N. Bennani and D. A. Menasce, Resource allocation for autonomic data centers using analytic performance models, in ICAC 2005.
- **[Bowm07]** I. T. Bowman, P. Bumbulis, D. Farrar, A. K. Goel, B. Lucier, A. Nica, G.N. Paulley, J. Smirios, M. Young-Lai. SQL Anywhere: A Holistic Approach to Database Self-management. SMDb 2007.

References

- **[brbr99]** John L. Bruno, Jose Carlos Brustoloni, Eran Gabber, Banu Ozden, and Abraham Silberschatz. Disk scheduling with quality of service guarantees. In IEEE International Conference on Multimedia Computing and Systems (ICMCS 1999), Vol. 2, pages 400-405, 1999.
- **[brch05]** Nicolas Bruno and Surajit Chaudhuri. Automatic physical database tuning: A relaxation-based approach. In Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data (SIGMOD'05), 2005.
- **[Brown93]** Brown, K.P., Carey, M.J., Livny, M. Managing memory to meet multiclass workload response time goals. In VLDB 1993.
- **[Brown96]** Brown, K.P., Carey, M.J., Livny, M. Goal-oriented buffer management revisited. In SIGMOD 1996.
- **[Chen06]** J. Chen, G. Soundararajan, C. Amza. Autonomic Provisioning of Backend Databases in Dynamic Content Web Servers. In ICAC 2006.
- **[Chen07]** J. Chen, G. Soundararajan, M. Mihailescu and C. Amza. Outlier Detection for Fine-grained Load Balancing in Database Clusters . In the 2nd International Workshop on Self-Managing Database Systems (SMDb) 2007.
- **[chzh05]** Z. Chen, Y. Zhang, Y. Zhou, H. Scott, and B. Schiefer. Empirical evaluation of multi-level buffer cache collaboration for storage systems. In Proceedings of the International Conference on Measurements and Modeling of Computer Systems (SIGMETRICS'05), pages 145-156, 2005.

References

- **[clfr05]** Christopher Clark, Keir Fraser, Steven Hand, Jacob Gorm Hansen, Eric Jul, Christian Limpach, Ian Pratt and Andrew Warfield. Live Migration of Virtual Machines. USENIX Symposium on Networked Systems Design and Implementation (NSDI), May 2005.
- **[coba05]** Mariano P. Consens, Denilson Barbosa, Adrian M. Teisanu, and Laurent Mignet. Goals and benchmarks for autonomic configuration recommenders. In Proc. ACM SIGMOD International Conference on Management of Data (SIGMOD'05), 2005.
- **[Da06]** K. Daudjee, K. Salem. Lazy database replication with snapshot isolation. In VLDB'06.
- **[Das05]** K.Dasgupta, S.Ghosal, R.Jain, U.Sharma, A.Verma, QoS Mig: Adaptive Rate-Controlled Migration of Bulk Data in Storage Systems, In *ICDE'05*.
- **[dech03]** Murthy Devarakonda, David Chess, Ian Whalley, Alla Segal, Pawan Goyal, Aamer Sachedina, Keri Romanufa, Ed Lassetre, William Tetzlaff, and Bill Arnold. Policy-based autonomic storage allocation. In Proc. 14th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM), number 2867 in Lecture Notes in Computer Science, pages 143-154. Springer-Verlag, 2003.

References

- **[Diao02]** Y. Diao, J. L. Hellerstein, AND S. Parekh, Optimizing quality of service using fuzzy control. In *Proceedings of the 13th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management, 2002*.
- **[Elni07]** S. Elnikety, S. Dropsho and W. Zwaenepoel. Tashkent+: Memory-Aware Load Balancing and Update Filtering in Replicated Databases. In EuroSys 2007.
- **[engu95]** Dawson R. Engler, Sandeep K. Gupta, and M. Frans Kaashoek. AVM: Application-level Virtual Memory. Workshop on Hot Topics in Operating Systems (HotOS), May 1995.
- **[fidi05]** Renato Figueiredo, Peter A Dinda, Jose Fortes. Resource Virtualization Renaissance. IEEE Computer, 38(5):28-31, May 2005.
- **[Ghan07]** S. Ghanbari, G. Soundararajan, J. Chen, and C. Amza Adaptive Learning of Metric Correlations for Temperature-Aware Database Provisioning, In ICAC 2007.
- **[goja03]** Pawan Goyal, Divyesh Jadav, Dharmendra S. Modha, and Renu Tewari. CacheCOW: QoS for storage system caches. In Eleventh International Workshop on Quality of Service (IWQoS 03), 2003.

References

- **[gold74]** Robert P. Goldberg. Survey of Virtual Machine Research. IEEE Computer, pp. 34-45, June 1974
- **[hape04]** Lan Huang, Gang Peng, and Tzi-cker Chiueh. Multi-dimensional storage virtualization. In Proc. Joint International Conference on Measurement and Modeling of Computer Systems, pages 14-24, 2004.
- **[kaka05]** Magnus Karlsson, Christos T. Karamanolis, and Xiaoyun Zhu. Triage: Performance differentiation for storage systems using adaptive control. ACM Transactions on Storage, 1(4):457-480, November 2005.
- **[Kar06]** A. Karve, T. Kimbrel, G. Pacifici, M. Spreitzer, M. Steinder, M. Sviridenko, A. Tantawi. Dynamic placement for clustered web applications. In *WWW 2006*.
- **[Kemme01]** B. Kemme, A. Bartoli, Ö. Babaoglu. Online Reconfiguration in Replicated Databases Based on Group Communication. In *DSN 2001*.
- **[kidu03]** Samuel T. King, George W. Dunlap, Peter M. Chen. Operating System Support for Virtual Machines. USENIX Annual Technical Conference, June 2003.
- **[Lah01]** T. Lahiri, V. Srihari, W. Chan, N. MacNaughton, and S. Chandrasekaran. Cache fusion: Extending shared-disk clusters with shared caches. In *VLDB 2001*.

References

- **[liab05]** X. Li, A. Aboulmaga, A. Sachedina, K. Salem, and S. Gao. Second-tier cache management using write hints. In USENIX Conference on File and Storage Technologies (FAST'05), pages 115-128, December 2005.
- **[liiy05]** Q. Lin, B. R. Iyer, D. Agrawal, and A. El Abbadi. SVL: Storage virtualization engine leveraging DBMS technology. In Proceedings of the 21st International Conference on Data Engineering (ICDE'05), pages 1048-1059, 2005.
- **[Lin05]** Y. Lin, B. Kemme, M. Patino-Martinez, and R. Jimenez-Peris. Middleware based Data Replication providing Snapshot Isolation, In *SIGMOD'05*.
- **[lowe98]** David B. Lomet and Gerhard Weikum. Efficient and Transparent Application Recovery in Client-Server Information Systems. SIGMOD Conference, pp. 460-471, June 1998.
- **[lume03]** Christopher Lumb, Arif Merchant, and Guillermo Alvarez. Facade: virtual storage devices with performance guarantees. In Proceedings of the 2nd USENIX Conference on File and Storage Technologies, pages 131-144, 2003.
- **[Mart06]** P. Martin, W. Powley, X. Xu, W. Tian. Automated Configuration of Multiple Buffer Pools. The Computer Journal, 2006.

References

- **[Mena01]** D. A. Menasce, D. Barbara, AND R. Dodge. Preserving QoS of e-commerce sites through self-tuning: A performance model approach. In Proceedings of the 3rd ACM Conference on Electronic Commerce, 2001.
- **[ozsa07]** O. Ozmen, K. Salem, M. Uysal and M. H. Sheikh Attar. Storage Workload Estimation for Database Management Systems. In Proc. ACM SIGMOD International Conference on Management of Data (SIGMOD'07), 2007.
- **[pazh07]** Pradeep Padala, Xiaoyun Zhu, Mustafa Uysal, Zhikui Wang, Sharad Singhal, Arif Merchant, and Kenneth Salem. Adaptive Control of Virtualized Resources in Utility Computing Environments. European Conference on Computer Systems, (EuroSys), March 2007.
- **[Peri02]** R. Jiménez-Peris, M. Patiño-Martínez, B. Kemme AND G. Alonso. Improving Scalability of Fault Tolerant Database Clusters. In *ICDCS'02*.
- **[Platt04]** C. Plattner and G. Alonso. Ganymed: Scalable replication for transactional web applications. In *Middleware 2004*.
- **[pogo74]** Gerald J. Popek and Robert P. Goldberg. Formal Requirements for Virtualizable Third Generation Architectures. Communications of the ACM, 17 (7): 412 –421, July 1974.

References

- **[Qin07]** Ye Qin, Kenneth Salem, and Anil Goel. Towards adaptive costing of database access methods. In *SMDB 2007*.
- **[roga05]** Mendel Rosenblum and Tal Garfinkel. Virtual Machine Monitors: Current Technology and Future Trends. *IEEE Computer*, 38(5):39-47, May 2005.
- **[roir00]** John Scott Robin and Cynthia E. Irvine. Analysis of the Intel Pentium's Ability to Support a Secure Virtual Machine Monitor. *USE*
- **[smna05]** James E. Smith and Ravi Nair. The Architecture of Virtual Machines. *IEEE Computer*, 38(5):32-38, May 2005.
- **[So06]** G. Soundararajan, C. Amza, A. Goel. Database Replication Policies for Dynamic Content Applications. In *EuroSys 2006*.
- **[So06]** G. Soundararajan, C. Amza. Reactive provisioning of backend databases in shared dynamic content server clusters. In *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, p151 – p188, Vol. 1, Issue 2 (December 2006).
- **[soab07]** Ahmed A. Soror, Ashraf Aboulnaga, and Kenneth Salem. Database Virtualization: A New Frontier for Database Tuning and Physical Design. International Workshop on Self-Managing Database Systems (*SMDB 2007*), April 2007.

References

- **[suve01]** Jeremy Sugerman, Ganesh Venkitachalam, and Beng-Hong Lim. Virtualizing I/O Devices on VMware Workstation's Hosted Virtual Machine Monitor. USENIX Annual Technical Conference, June 2001.
- **[Tes05]** G. Tesauro, R. Das, W. E. Walsh, and J. O. Kephart. Utility-function-driven resource allocation. in ICAC 2005.
- **[Tes06]** G. Tesauro, R. Das, N. Jong, M. Bennani. A hybrid reinforcement learning approach to autonomic resource allocation. In ICAC 2006.
- **[Tot06]** A. Totok and V. Karamcheti. Improving performance of internet services through reward-driven request prioritization. In IWQoS 2006.
- **[Urga05]** B. Urgaonkar, G. Pacifici, P. Shenoy, M. Spreitzer and A. Tantawi. An analytical model for multi-tier internet services and its applications. In SIGMETRICS 2005
- **[utyi05]** Sandeep Uttamchandani, Li Yin, Guillermo A. Alvarez, John Palmer, and Gul Agha. CHAMELEON: A self-evolving, fully-adaptive resource arbitrator for storage systems. In Proc. USENIX 2005 Annual Technical Conference, pages 75-88, 2005.
- **[vazu00]** Gary Valentin, Michael Zuliani, Daniel C. Zilio, Guy M. Lohman, and Alan Skelley. DB2 Advisor: An optimizer smart enough to recommend its own indexes. In 16th International Conference on Data Engineering, pages 101-110, 2000.

References

- **[vome04]** K. Voruganti, J. Menon, and S. Gopisetty. Land below a DBMS. SIGMOD Record, 33(1):64-70, March 2004.
- **[wal02]** Carl A. Waldspurger. Memory Resource Management in VMware ESX Server. Symposium on Operating Systems Design and Implementation (OSDI), December 2002.
- **[Wal04]** W.E. Walsh, G. Tesauro, J. O. Kephart, and R. Das. Utility functions in autonomic systems. In ICAC 2004.
- **[waos02]** Julie Ward, Michael O'Sullivan, Troy Shahoumian, and John Wilkes. Appia: automatic storage area network design. In Conference on File and Storage Technology (FAST'02), pages 203-217, January 2002.
- **[wech04]** Wei Jin, Jeffrey S. Chase, and Jasleen Kaur. Interposed proportional sharing for a storage service utility. In Proc. International Conference on Measurements and Modeling of Computer Systems (SIGMETRICS'04), pages 37-48, June 2004.
- **[whco05]** Andrew Whitaker, Richard S. Cox, Marianne Shaw, and Steven D. Gribble. Rethinking the Design of Virtual Machine Monitors. IEEE Computer, 38(5):57-62, May 2005.

References

- **[whsh02]** Andrew Whitaker, Marianne Shaw, and Steven D. Gribble. Denali: A Scalable Isolation Kernel. ACM SIGOPS European Workshop, September 2002.
- **[Wood06]** M. Woodside, T. Zheng, M. Litoiu. Service System Resource Management Based on a Tracked Layered Performance Model. In *ICAC 2006*.