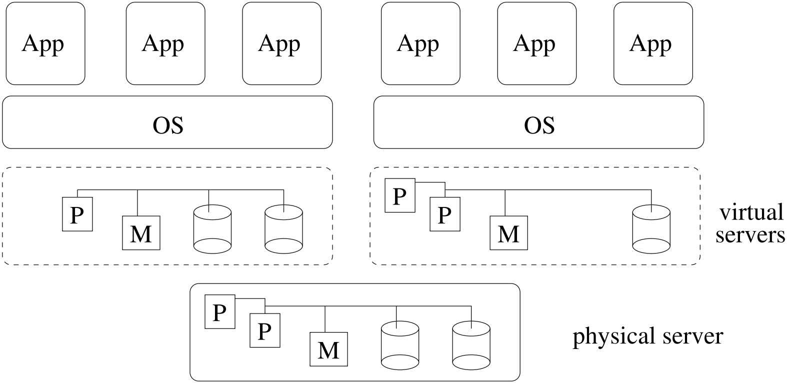
CS848 Management of Information Systems
Fall 2006

Ken Salem

David R. Cheriton School of Computer Science
University of Waterloo

Virtualization



Server Virtualization

Server virtualization adds a layer of indirection between the OS and the physical server hardware.

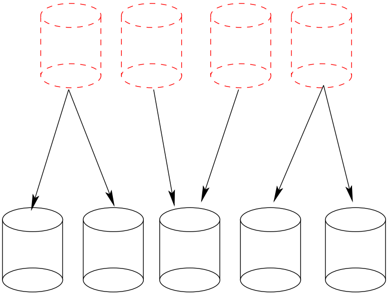
Virtual Machine Monitors

- VMMs implement virtual servers
- Examples:
  - Xen
  - VMWare (various products)
- VMM administrative interface allows virtual machines to be configured, tuned, and controlled
  - virtual hardware configuration, e.g., number of processors, memory size, disks, network interface
  - power on/off, suspend/resume, checkpoint
  - dynamic tuning, e.g., CPU scheduling parameters, memory size

Why Virtualize?

- resource management, resource consolidation
  - replace silo deployments with shared data center
- fault isolation, security
  - VMM provides better isolation than OS
- application compatibility
- support software lifecycle
  - heterogeneous testing environments, duplicate environments
- software deployment

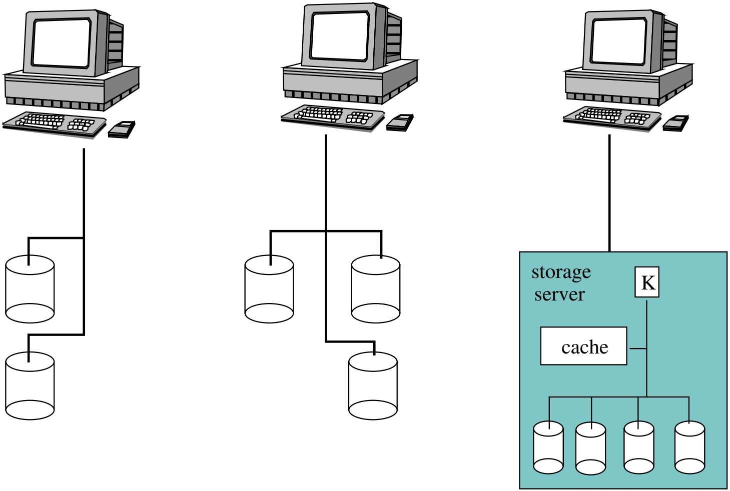
Storage Virtualization



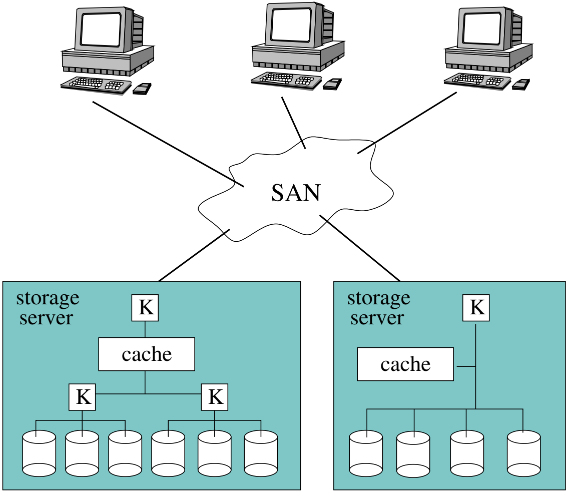
- configuration and adjustment of virtual device characteristics
  - capacity
  - performance
  - reliability
- management automation
  - dynamic allocation
  - hierarchical storage management

Storage virtualization adds indirection above the physical hardware.

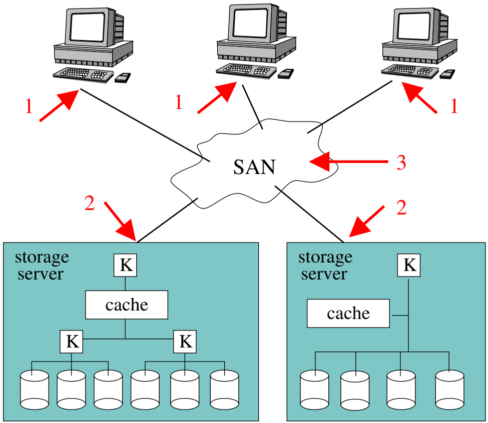
Storage Architecture: DAS



Storage Architecture: SAN



Storage Virtualization Points



1. **storage virtualization at the host**
2. **storage virtualization at the target**
3. **storage virtualization in the SAN**

Control Objectives

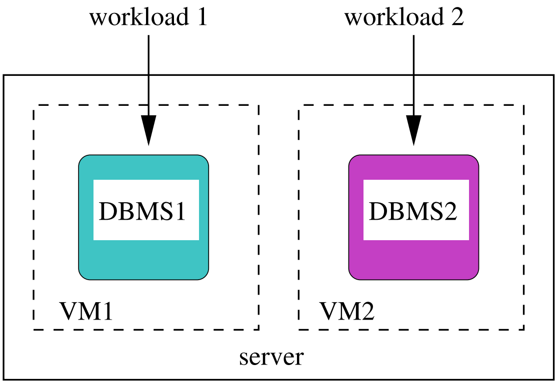
regulation: adjust control input so that measured output stays close to a reference value.

optimization: adjust control input to optimize (maximize or minimize) a measured output.

Challenges

Characteristics of the target system, characteristics of the workload, and the reference value may all change over time.

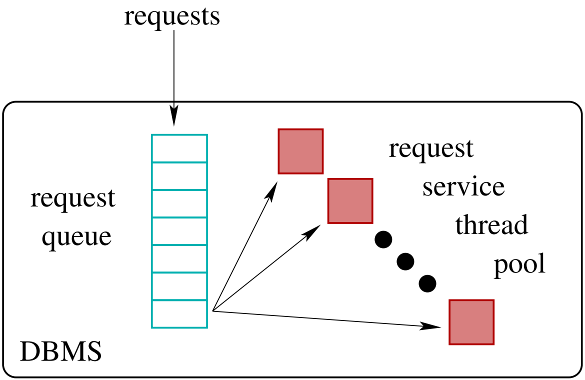
Regulation Example



Regulation Problem

Adjust the CPU share of VM1 to maintain a mean request response time for DBMS1 of 2 seconds.

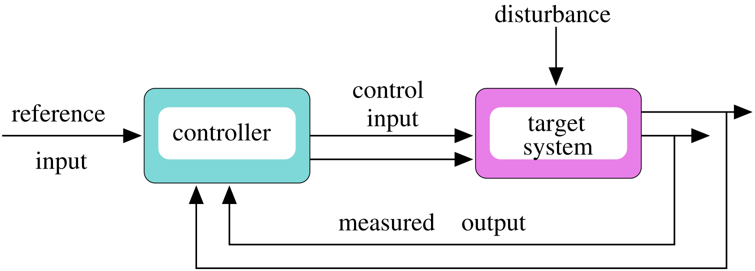
Optimization Example



Optimization Problem

Adjust size of request service thread pool to maximize the request throughput.

Feedback (Closed-Loop) Control

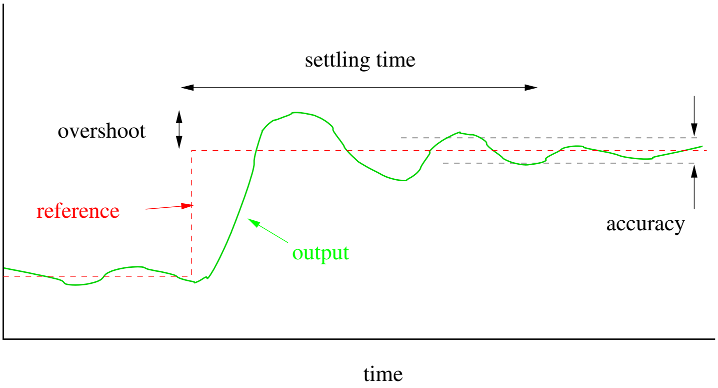


SISO control: single input, single output

MISO control: multi-input, single output

MIMO control: multi-input, multi-output

Controller Properties



Desirable Controller Properties

Stability, accuracy, small setting time, small overshoot

Basic Controllers: Proportional (P) Control

- control rule: $u(t) = K_P e(t)$
- control input is proportional to error
- inaccurate but fast
  - control input moves quickly in response to errors
  - $e(t) = 0$ implies $u(t) = 0$

Notation

$u(t)$: control input at time t

$r(t)$: reference input at time t

$y(t)$: measured output at time t

$e(t) = r(t) - y(t)$: control error at time t

Basic Controllers: Integral (I) Control

- control rule: $u(t) = u(t-1) + K_I e(t)$
- change in control input is proportional to error
- slow but accurate

Notation

$u(t)$: control input at time t

$r(t)$: reference input at time t

$y(t)$: measured output at time t

$e(t) = r(t) - y(t)$: control error at time t

Basic Controllers: Derivative (D) Control

- control rule: $u(t) = K_D[e(t) - e(t-1)]$
- control input is proportional to change in error
- idea: react quickly to large changes in error

Notation

$u(t)$: control input at time t

$r(t)$: reference input at time t

$y(t)$: measured output at time t

$e(t) = r(t) - y(t)$: control error at time t

More Controllers

- controllers can be combined to balance strengths and weaknesses
- example: PI control combines P control and I control
  - $u(t) = u(t-1) + (K_P + K_I)e(t) - K_P e(t-1)$
- other examples: PD control, PID control

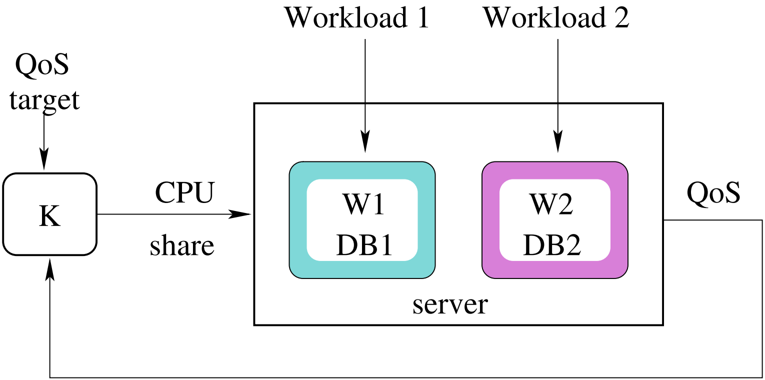
Controller Design

- the controller design problem is the problem of designing a control rule for a particular target system
- controller designs seek to ensure stability and to achieve a balance among accuracy, settling time, and overshoot
- in the case of P, I, and D controllers, controller design amounts to choosing values for the constants K_P, K_I, K_D.

Target System Modeling

Controller design depends on having a model of the target system. A model relates the control input(s) $u(t)$ to the measured output(s) $y(t)$.

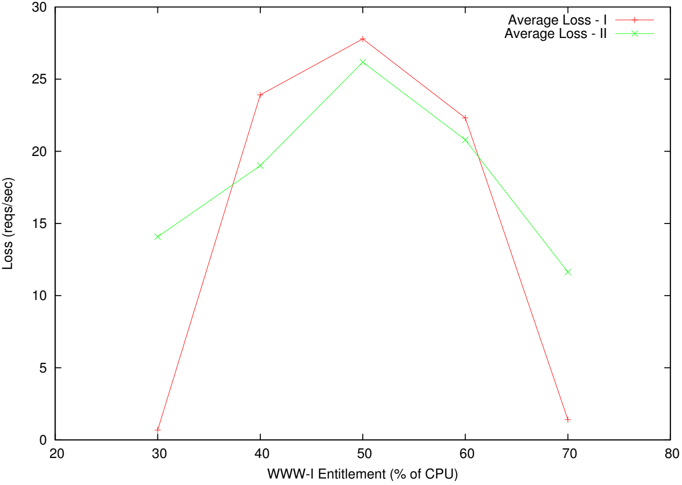
Control Example



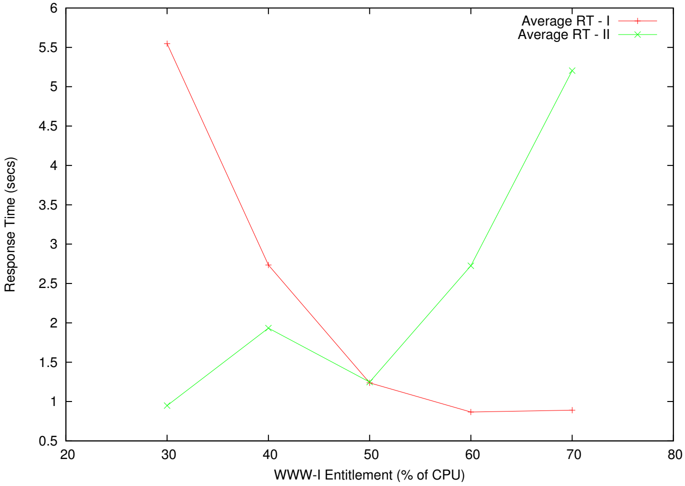
QoS Metrics

Possible QoS outputs include request loss ratio and response time ratio.

Control Example: Measured Loss Ratios



Control Example: Measured Response Times



Control Example: Measured Ratios