QoS Optimization

Multiple QoS Levels

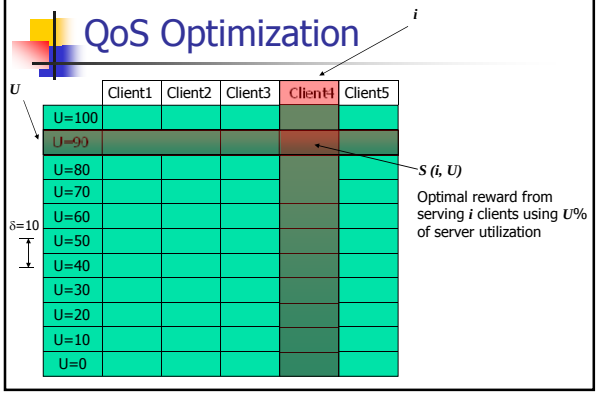
QoS Adaptation

- Notion of QoS levels
  - Video server can serve movies at three resolutions:
    - Level1: 40% CPU, 100% Utility
    - Level2: 30% CPU, 70% Utility
    - Level3: 10% CPU, 10% Utility
  - Same server can broadcast news at three resolutions:
    - Level1: 30% CPU, 100% Utility
    - Level2: 20% CPU, 90% Utility
    - Level3: 10% CPU, 60% Utility
- Customers pay
  - $1/hr of movie streaming
  - 65 cents/hr of news streaming
- Degraded QoS levels are discounted proportionally to their utility
- How to maximize server revenue?

QoS Optimization

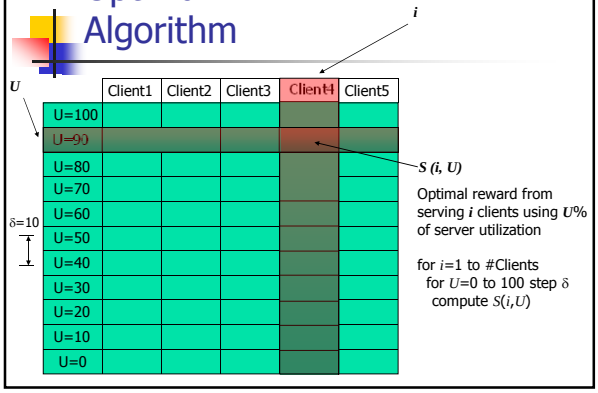
	Client1	Client2	Client3	Client4	Client5
U=100					
U=90					
U=80					
U=70					
U=60					
U=50					
U=40					
U=30					
U=20					
U=10					
U=0					

QoS Optimization



Optimal reward from serving ***i*** clients using ***U***% of server utilization

Optimal Algorithm

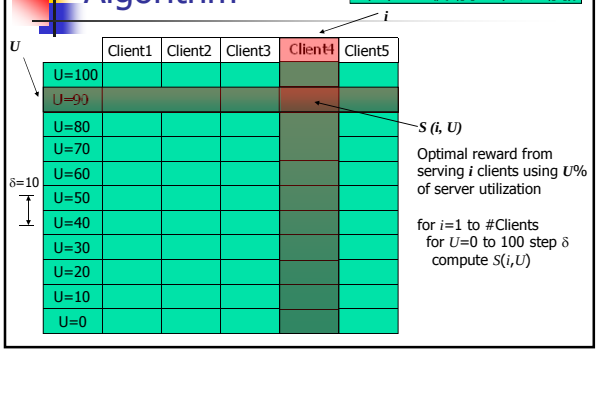


Optimal reward from serving ***i*** clients using ***U***% of server utilization

for i=1 to #Clients
  for U=0 to 100 step δ
    compute $S(i,U)$

Optimal Algorithm

$R_i[k]$: Reward for serving client i at QoS level k
$U_i[k]$: utilization of QoS level k of client i
$S(i,U) = \max_k \{R_i[k] + S(i\text{-}1, U\text{-}U_i[k])\}$



Optimal reward from serving ***i*** clients using ***U***% of server utilization

for i=1 to #Clients
  for U=0 to 100 step δ
    compute $S(i,U)$

QoS Optimization Example

- Example
  - Client 1, 2:
    - Level1: 40% CPU, $1
    - Level2: 30% CPU, $0.7
    - Level3: 10% CPU, $0.1
  - Client 3, 4:
    - Level1: 30% CPU, $0.65
    - Level2: 20% CPU, $0.6
    - Level3: 10% CPU, $0.4

QoS Optimization Example

- Example (i)
  - Client 1, 2:
    - (k) Level1: 40% CPU, $1
    - Level2: 30% CPU, $0.7
    - Level3: 10% CPU, $0.1
  - Client 3, 4:
    - Level1: 30% CPU, $0.65
    - Level2: 20% CPU, $0.6
    - Level3: 10% CPU, $0.4

$U_i[k]$ $R_i[k]$

Example

$R_i[k]$: Reward for serving client i at QoS level k
$U_i[k]$: utilization of QoS level k of client i
$S(i,U) = \max_k \{R_i[k] + S(i\text{-}1, U\text{-}U_i[k])\}$

i

U

	Client1	Client2	Client3	Client4
U=100				
U=90				
U=80				
U=70				
U=60				
U=50				
U=40				
U=30				
U=20				
U=10				
U=0				

δ=10

Client 1, 2:
Level1: 40% CPU, $1
Level2: 30% CPU, $0.7
Level3: 10% CPU, $0.1
Client 3, 4:
Level1: 30% CPU, $0.65
Level2: 20% CPU, $0.6
Level3: 10% CPU, $0.4

$U_i[k]$ $R_i[k]$

Example

$R_i[k]$: Reward for serving client i at QoS level k
$U_i[k]$: utilization of QoS level k of client i
$S(i,U) = \max_k \{R_i[k] + S(i\text{-}1,U\text{-}U_i[k])\}$

i

U

	Client1	Client2	Client3	Client4
U=100	1(1)			
U=90	1(1)			
U=80	1(1)			
U=70	1(1)			
U=60	1(1)			
U=50	1(1)			
U=40	1(1)			
U=30	0.7(2)			
U=20	0.1(3)			
U=10	0.1(3)			
U=0	0			

δ=10

Client 1, 2:
Level1: 40% CPU, $1
Level2: 30% CPU, $0.7
Level3: 10% CPU, $0.1
Client 3, 4:
Level1: 30% CPU, $0.65
Level2: 20% CPU, $0.6
Level3: 10% CPU, $0.4

$U_i[k]$ $R_i[k]$

Example

$R_i[k]$: Reward for serving client i at QoS level k
$U_i[k]$: utilization of QoS level k of client i
$S(i,U) = \max_k \{R_i[k] + S(i\text{-}1,U\text{-}U_i[k])\}$

i

U

	Client1	Client2	Client3	Client4
U=100	1(1)	2(1)		
U=90	1(1)	2(1)		
U=80	1(1)	2(1)		
U=70	1(1)	1.7(1)		
U=60	1(1)	1.4(1)		
U=50	1(1)	1.1(1)		
U=40	1(1)	1(1)		
U=30	0.7(2)	0.7(2)		
U=20	0.1(3)	0.2(3)		
U=10	0.1(3)	0.1(3)		
U=0	0	0		

δ=10

Client 1, 2:
Level1: 40% CPU, $1
Level2: 30% CPU, $0.7
Level3: 10% CPU, $0.1
Client 3, 4:
Level1: 30% CPU, $0.65
Level2: 20% CPU, $0.6
Level3: 10% CPU, $0.4

$U_i[k]$ $R_i[k]$

Example

$R_i[k]$: Reward for serving client i at QoS level k
$U_i[k]$: utilization of QoS level k of client i
$S(i,U) = \max_k \{R_i[k] + S(i\text{-}1,U\text{-}U_i[k])\}$

i

U

	Client1	Client2	Client3	Client4
U=100	1(1)	2(1)	2.6(2)	
U=90	1(1)	2(1)	2.4(3)	
U=80	1(1)	2(1)	2.1(3)	
U=70	1(1)	1.7(1)	1.8(3)	
U=60	1(1)	1.4(1)	1.6(2)	
U=50	1(1)	1.1(1)	1.4(3)	
U=40	1(1)	1(1)	1.1(3)	
U=30	0.7(2)	0.7(2)	0.7(2)	
U=20	0.1(3)	0.2(3)	0.6(2)	
U=10	0.1(3)	0.1(3)	0.4(3)	
U=0	0	0	0	

δ=10

Client 1, 2:
Level1: 40% CPU, $1
Level2: 30% CPU, $0.7
Level3: 10% CPU, $0.1
Client 3, 4:
Level1: 30% CPU, $0.65
Level2: 20% CPU, $0.6
Level3: 10% CPU, $0.4

$U_i[k]$ $R_i[k]$

Example

$R_i[k]$: Reward for serving client i at QoS level k
$U_i[k]$: utilization of QoS level k of client i
$S(i,U) = \max_k \{R_i[k] + S(i\text{-}1, U\text{-}U_i[k])\}$

i

U	Client1	Client2	Client3	Client4
U=100	1(1)	2(1)	2.6(2)	2.8(3)
U=90	1(1)	2(1)	2.4(3)	2.5(3)
U=80	1(1)	2(1)	2.1(3)	2.2(2)
U=70	1(1)	1.7(1)	1.8(3)	2(2)
U=60	1(1)	1.4(1)	1.6(2)	1.8(3)
U=50	1(1)	1.1(1)	1.4(3)	1.5(3)
U=40	1(1)	1(1)	1.1(3)	1.2(2)
U=30	0.7(2)	0.7(2)	0.7(2)	1(2)
U=20	0.1(3)	0.2(3)	0.6(2)	0.8(3)
U=10	0.1(3)	0.1(3)	0.4(3)	0.4(3)
U=0	0	0	0	0

δ=10

Client 1, 2:
Level1: 40% CPU, $1
Level2: 30% CPU, $0.7
Level3: 10% CPU, $0.1
Client 3, 4:
Level1: 30% CPU, $0.65
Level2: 20% CPU, $0.6
Level3: 10% CPU, $0.4

$U_i[k]$ $R_i[k]$

Solution

$R_i[k]$: Reward for serving client i at QoS level k
$U_i[k]$: utilization of QoS level k of client i
$S(i,U) = \max_k \{R_i[k] + S(i\text{-}1, U\text{-}U_i[k])\}$

i

U	Client1	Client2	Client3	Client4
U=100	1(1)	2(1)	2.6(2)	**2.8(3)**
U=90	1(1)	2(1)	**2.4(3)**	2.5(3)
U=80	1(1)	**2(1)**	2.1(3)	2.2(2)
U=70	1(1)	1.7(1)	1.8(3)	2(2)
U=60	1(1)	1.4(1)	1.6(2)	1.8(3)
U=50	1(1)	1.1(1)	1.4(3)	1.5(3)
U=40	**1(1)**	1(1)	1.1(3)	1.2(2)
U=30	0.7(2)	0.7(2)	0.7(2)	1(2)
U=20	0.1(3)	0.2(3)	0.6(2)	0.8(3)
U=10	0.1(3)	0.1(3)	0.4(3)	0.4(3)
U=0	0	0	0	0

δ=10

Client 1, 2:
Level1: 40% CPU, $1
Level2: 30% CPU, $0.7
Level3: 10% CPU, $0.1
Client 3, 4:
Level1: 30% CPU, $0.65
Level2: 20% CPU, $0.6
Level3: 10% CPU, $0.4

$U_i[k]$ $R_i[k]$

Approximate QoS Optimization

Hill Climbing Algorithm

How to get quickly to the top and stay there longest?

Approximate QoS Optimization

Hill Climbing Algorithm

How to get quickly to the top
and stay there longest?

- Ascend the path of maximum slope
- Descend the path of minimum slope

Approximate QoS Optimization

Hill Climbing Algorithm

How to get quickly to the top
and stay there longest?

- Ascend the path of maximum slope
- Descend the path of minimum slope

Loop
  if underutilized then take maximum slope promotion
  if overload then take minimum slope demotion
End Loop

Approximate QoS Optimization

Hill Climbing Algorithm

How to get quickly to the top
and stay there longest?

- Ascend the path of maximum slope
- Descend the path of minimum slope

$$\text{Slope} = \frac{R_i[new] - R_i[old]}{U_i[new] - U_i[old]}$$

Loop
  if underutilized then take maximum slope promotion
  if overload then take minimum slope demotion
End Loop

Approximate QoS Optimization

Hill Climbing Algorithm

Client 1, 2:
Level1: 40% CPU, $1
Level2: 30% CPU, $0.7
Level3: 10% CPU, $0.1
Client 3, 4:
Level1: 30% CPU, $0.65
Level2: 20% CPU, $0.6
Level3: 10% CPU, $0.4

Slope = $\frac{R_i[new] - R_i[old]}{U_i[new] - U_i[old]}$

0% 0,0,0,0

Loop
if underutilized then take maximum slope promotion
if overload then take minimum slope demotion
End Loop

Approximate QoS Optimization

Hill Climbing Algorithm

Client 1, 2:
Level1: 40% CPU, $1
Level2: 30% CPU, $0.7
Level3: 10% CPU, $0.1
Client 3, 4:
Level1: 30% CPU, $0.65
Level2: 20% CPU, $0.6
Level3: 10% CPU, $0.4

Promote 3 or 4 to L3: Slope=0.4/0.1=4
Promote 1 or 2 to L1: Slope=1/0.4=2.5
Promote 3 or 4 to L2: Slope=0.6/0.2=3
Promote 1 or 2 to L2: Slope=0.7/0.3=2.33
Promote 3 or 4 to L1: Slope=0.65/0.3=2.02
Promote 1 or 2 to L3: Slope=0.1/0.1=1

Slope = $\frac{R_i[new] - R_i[old]}{U_i[new] - U_i[old]}$

0% 0,0,0,0

Loop
if underutilized then take maximum slope promotion
if overload then take minimum slope demotion
End Loop

Approximate QoS Optimization

Hill Climbing Algorithm

Client 1, 2:
Level1: 40% CPU, $1
Level2: 30% CPU, $0.7
Level3: 10% CPU, $0.1
Client 3, 4:
Level1: 30% CPU, $0.65
Level2: 20% CPU, $0.6
Level3: 10% CPU, $0.4

Promote 3 or 4 to L3: Slope=0.4/0.1=**4**
Promote 1 or 2 to L1: Slope=1/0.4=2.5
Promote 3 or 4 to L2: Slope=0.6/0.2=3
Promote 1 or 2 to L2: Slope=0.7/0.3=2.33
Promote 3 or 4 to L1: Slope=0.65/0.3=2.02
Promote 1 or 2 to L3: Slope=0.1/0.1=1

Slope = $\frac{R_i[new] - R_i[old]}{U_i[new] - U_i[old]}$

0% 0,0,0,0

Loop
if underutilized then take maximum slope promotion
if overload then take minimum slope demotion
End Loop

Approximate QoS Optimization

Hill Climbing Algorithm

Client 1, 2:
Level1: 40% CPU, $1
Level2: 30% CPU, $0.7
Level3: 10% CPU, $0.1
Client 3, 4:
Level1: 30% CPU, $0.65
Level2: 20% CPU, $0.6
Level3: 10% CPU, $0.4

Slope = $\frac{R_i[new] - R_i[old]}{U_i[new] - U_i[old]}$

10% 0,0,L3,0

Promote 3 to L3:
Slope=0.4/0.1=**4**

0% 0,0,0,0

Loop
if underutilized then take maximum slope promotion
if overload then take minimum slope demotion
End Loop

Approximate QoS Optimization

Hill Climbing Algorithm

Client 1, 2:
Level1: 40% CPU, $1
Level2: 30% CPU, $0.7
Level3: 10% CPU, $0.1
Client 3, 4:
Level1: 30% CPU, $0.65
Level2: 20% CPU, $0.6
Level3: 10% CPU, $0.4

Promote 4 to L3:
Slope=0.4/0.1=**4**

Promote 1 or 2 to L1:
Slope=1/0.4=2.5

Promote 4 to L2:
Slope=0.6/0.2=3

Promote 1 or 2 to L2:
Slope=0.7/0.3=2.33

Promote 4 to L1:
Slope=0.65/0.3=2.02

Promote 1 or 2 to L3:
Slope=0.1/0.1=1

Slope = $\frac{R_i[new] - R_i[old]}{U_i[new] - U_i[old]}$

10% 0,0,L3,0

Promote 3 to L3:
Slope=0.4/0.1=**4**

0% 0,0,0,0

Loop
if underutilized then take maximum slope promotion
if overload then take minimum slope demotion
End Loop

Approximate QoS Optimization

Hill Climbing Algorithm

Client 1, 2:
Level1: 40% CPU, $1
Level2: 30% CPU, $0.7
Level3: 10% CPU, $0.1
Client 3, 4:
Level1: 30% CPU, $0.65
Level2: 20% CPU, $0.6
Level3: 10% CPU, $0.4

Promote 1 or 2 to L1:
Slope=1/0.4=**2.5**

Promote 1 or 2 to L2:
Slope=0.7/0.3=2.33

Promote 1 or 2 to L3:
Slope=0.1/0.1=1

Slope = $\frac{R_i[new] - R_i[old]}{U_i[new] - U_i[old]}$

20% 0,0,L3,L3

Promote 4 to L3:
Slope=0.4/0.1=**4**

10% 0,0,L3,0

Promote 3 to L3:
Slope=0.4/0.1=**4**

0% 0,0,0,0

Loop
if underutilized then take maximum slope promotion
if overload then take minimum slope demotion
End Loop

Approximate QoS Optimization

Hill Climbing Algorithm

Client 1, 2:
Level1: 40% CPU, $1
Level2: 30% CPU, $0.7
Level3: 10% CPU, $0.1
Client 3, 4:
Level1: 30% CPU, $0.65
Level2: 20% CPU, $0.6
Level3: 10% CPU, $0.4

60% L1,0,L3,L3
Promote 1 to L1: Slope=1/0.4=**2.5**
20% 0,0,L3,L3
Promote 4 to L3: Slope=0.4/0.1=**4**
10% 0,0,L3,0
Promote 3 to L3: Slope=0.4/0.1=**4**
0% 0,0,0,0

Slope = $\frac{R_i[new] - R_i[old]}{U_i[new] - U_i[old]}$

Loop
if underutilized then take maximum slope promotion
if overload then take minimum slope demotion
End Loop

Approximate QoS Optimization

Hill Climbing Algorithm

Client 1, 2:
Level1: 40% CPU, $1
Level2: 30% CPU, $0.7
Level3: 10% CPU, $0.1
Client 3, 4:
Level1: 30% CPU, $0.65
Level2: 20% CPU, $0.6
Level3: 10% CPU, $0.4

Promote 2 to L1: Slope=1/0.4=**2.5**
L1,L1,L3,L3
60% L1,0,L3,L3
Promote 1 to L1: Slope=1/0.4=**2.5**
20% 0,0,L3,L3
Promote 4 to L3: Slope=0.4/0.1=**4**
10% 0,0,L3,0
Promote 3 to L3: Slope=0.4/0.1=**4**
0% 0,0,0,0

Slope = $\frac{R_i[new] - R_i[old]}{U_i[new] - U_i[old]}$

Loop
if underutilized then take maximum slope promotion
if overload then take minimum slope demotion
End Loop

Approximate QoS Optimization

Hill Climbing Algorithm

Client 1, 2:
Level1: 40% CPU, $1
Level2: 30% CPU, $0.7
Level3: 10% CPU, $0.1
Client 3, 4:
Level1: 30% CPU, $0.65
Level2: 20% CPU, $0.6
Level3: 10% CPU, $0.4

Promote 2 to L1: Slope=1/0.4=**2.5**
L1,L1,L3,L3
Reward 2.8
60% L1,0,L3,L3
Promote 1 to L1: Slope=1/0.4=**2.5**
20% 0,0,L3,L3
Promote 4 to L3: Slope=0.4/0.1=**4**
10% 0,0,L3,0
Promote 3 to L3: Slope=0.4/0.1=**4**
0% 0,0,0,0

Slope = $\frac{R_i[new] - R_i[old]}{U_i[new] - U_i[old]}$

Loop
if underutilized then take maximum slope promotion
if overload then take minimum slope demotion
End Loop

Approximate QoS Optimization

Restricted Hill Climbing

Client 1, 2:
Level1: 40% CPU, $1
Level2: 30% CPU, $0.7
Level3: 10% CPU, $0.1
Client 3, 4:
Level1: 30% CPU, $0.65
Level2: 20% CPU, $0.6
Level3: 10% CPU, $0.4

Slope = $\frac{R_i[new] - R_i[old]}{U_i[new] - U_i[old]}$

0% 0,0,0,0

Loop
if underutilized then take maximum slope ***single level*** promotion
if overload then take minimum slope ***single level*** demotion
End Loop

Approximate QoS Optimization

Restricted Hill Climbing

Promote 3 or 4 to L2:
Slope=0.2/0.1=2

Promote 1 or 2 to L3:
Slope=0.1/0.1=1

20% 0,0,L3,L3

Promote 4 to L3:
Slope=0.4/0.1=**4**

10% 0,0,L3,0

Promote 3 to L3:
Slope=0.4/0.1=**4**

0% 0,0,0,0

Client 1, 2:
Level1: 40% CPU, $1
Level2: 30% CPU, $0.7
Level3: 10% CPU, $0.1
Client 3, 4:
Level1: 30% CPU, $0.65
Level2: 20% CPU, $0.6
Level3: 10% CPU, $0.4

Slope = $\frac{R_i[new] - R_i[old]}{U_i[new] - U_i[old]}$

Loop
if underutilized then take maximum slope ***single level*** promotion
if overload then take minimum slope ***single level*** demotion
End Loop

Approximate QoS Optimization

Restricted Hill Climbing

Promote 3 or 4 to L2:
Slope=0.2/0.1=2

Promote 1 or 2 to L3:
Slope=0.1/0.1=1

20% 0,0,L3,L3

Promote 4 to L3:
Slope=0.4/0.1=**4**

10% 0,0,L3,0

Promote 3 to L3:
Slope=0.4/0.1=**4**

0% 0,0,0,0

Client 1, 2:
Level1: 40% CPU, $1
Level2: 30% CPU, $0.7
Level3: 10% CPU, $0.1
Client 3, 4:
Level1: 30% CPU, $0.65
Level2: 20% CPU, $0.6
Level3: 10% CPU, $0.4

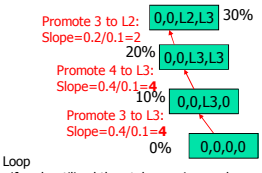
Slope = $\frac{R_i[new] - R_i[old]}{U_i[new] - U_i[old]}$

Loop
if underutilized then take maximum slope ***single level*** promotion
if overload then take minimum slope ***single level*** demotion
End Loop

Approximate QoS Optimization

Restricted Hill Climbing

Client 1, 2:
Level1: 40% CPU, $1
Level2: 30% CPU, $0.7
Level3: 10% CPU, $0.1
Client 3, 4:
Level1: 30% CPU, $0.65
Level2: 20% CPU, $0.6
Level3: 10% CPU, $0.4

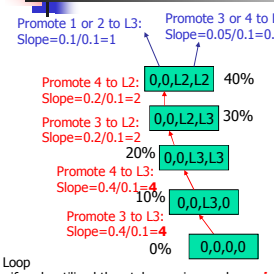


$$\text{Slope} = \frac{R_i[new] - R_i[old]}{U_i[new] - U_i[old]}$$

Loop
if underutilized then take maximum slope ***single level*** promotion
if overload then take minimum slope ***single level*** demotion
End Loop

Approximate QoS Optimization

Client 1, 2:
Level1: 40% CPU, $1
Level2: 30% CPU, $0.7
Level3: 10% CPU, $0.1
Client 3, 4:
Level1: 30% CPU, $0.65
Level2: 20% CPU, $0.6
Level3: 10% CPU, $0.4

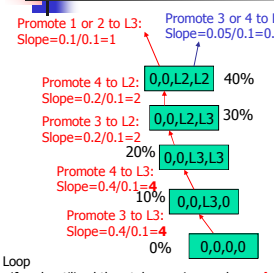


$$\text{Slope} = \frac{R_i[new] - R_i[old]}{U_i[new] - U_i[old]}$$

Loop
if underutilized then take maximum slope ***single level*** promotion
if overload then take minimum slope ***single level*** demotion
End Loop

Approximate QoS Optimization

Client 1, 2:
Level1: 40% CPU, $1
Level2: 30% CPU, $0.7
Level3: 10% CPU, $0.1
Client 3, 4:
Level1: 30% CPU, $0.65
Level2: 20% CPU, $0.6
Level3: 10% CPU, $0.4



$$\text{Slope} = \frac{R_i[new] - R_i[old]}{U_i[new] - U_i[old]}$$

Loop
if underutilized then take maximum slope ***single level*** promotion
if overload then take minimum slope ***single level*** demotion
End Loop

Approximate QoS Optimization

Client 1, 2:
- Level1: 40% CPU, $1
- Level2: 30% CPU, $0.7
- Level3: 10% CPU, $0.1

Client 3, 4:
- Level1: 30% CPU, $0.65
- Level2: 20% CPU, $0.6
- Level3: 10% CPU, $0.4

Slope = $\frac{R_i[new] - R_i[old]}{U_i[new] - U_i[old]}$

50% L3,0,L2,L2

Loop
if underutilized then take maximum slope ***single level*** promotion
if overload then take minimum slope ***single level*** demotion
End Loop

Approximate QoS Optimization

Client 1, 2:
- Level1: 40% CPU, $1
- Level2: 30% CPU, $0.7
- Level3: 10% CPU, $0.1

Client 3, 4:
- Level1: 30% CPU, $0.65
- Level2: 20% CPU, $0.6
- Level3: 10% CPU, $0.4

Slope = $\frac{R_i[new] - R_i[old]}{U_i[new] - U_i[old]}$

L2,0,L2,L2 70%

Promote 1 to L2:
Slope=0.6/0.2=3

50% L3,0,L2,L2

Loop
if underutilized then take maximum slope ***single level*** promotion
if overload then take minimum slope ***single level*** demotion
End Loop

Approximate QoS Optimization

Client 1, 2:
- Level1: 40% CPU, $1
- Level2: 30% CPU, $0.7
- Level3: 10% CPU, $0.1

Client 3, 4:
- Level1: 30% CPU, $0.65
- Level2: 20% CPU, $0.6
- Level3: 10% CPU, $0.4

Slope = $\frac{R_i[new] - R_i[old]}{U_i[new] - U_i[old]}$

L1,0,L2,L2 80%

Promote 1 to L1:
Slope=0.3/0.1=3

L2,0,L2,L2 70%

Promote 1 to L2:
Slope=0.6/0.2=3

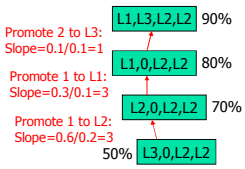
50% L3,0,L2,L2

Loop
if underutilized then take maximum slope ***single level*** promotion
if overload then take minimum slope ***single level*** demotion
End Loop

Approximate QoS Optimization

Client 1, 2:
- Level1: 40% CPU, $1
- Level2: 30% CPU, $0.7
- Level3: 10% CPU, $0.1

Client 3, 4:
- Level1: 30% CPU, $0.65
- Level2: 20% CPU, $0.6
- Level3: 10% CPU, $0.4



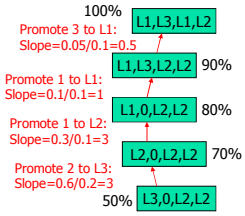
Slope = $\frac{R_i[new] - R_i[old]}{U_i[new] - U_i[old]}$

Loop
if underutilized then take maximum slope ***single level*** promotion
if overload then take minimum slope ***single level*** demotion
End Loop

Approximate QoS Optimization

Client 1, 2:
- Level1: 40% CPU, $1
- Level2: 30% CPU, $0.7
- Level3: 10% CPU, $0.1

Client 3, 4:
- Level1: 30% CPU, $0.65
- Level2: 20% CPU, $0.6
- Level3: 10% CPU, $0.4



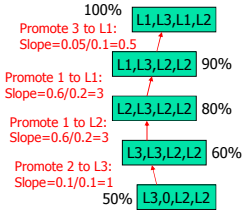
Slope = $\frac{R_i[new] - R_i[old]}{U_i[new] - U_i[old]}$

Loop
if underutilized then take maximum slope ***single level*** promotion
if overload then take minimum slope ***single level*** demotion
End Loop

Approximate QoS Optimization

Client 1, 2:
- Level1: 40% CPU, $1
- Level2: 30% CPU, $0.7
- Level3: 10% CPU, $0.1

Client 3, 4:
- Level1: 30% CPU, $0.65
- Level2: 20% CPU, $0.6
- Level3: 10% CPU, $0.4



Slope = $\frac{R_i[new] - R_i[old]}{U_i[new] - U_i[old]}$

Loop
if underutilized then take maximum slope ***single level*** promotion
if overload then take minimum slope ***single level*** demotion
End Loop