

vmworld2005

virtualize^{now}

las vegas • october 18-20, 2005

PAC485

Managing Datacenter Resources Using the VirtualCenter Distributed Resource Scheduler

Carl Waldspurger
Principal Engineer, R&D

**This presentation may contain VMware
confidential information.**

Copyright © 2005 VMware, Inc. All rights reserved. All other
marks and names mentioned herein may be trademarks of their respective
companies.

Talk Overview

- Context and features
- Managing resources
- Virtual machine placement
- System architecture
- Summary

What Is DRS?

- DRS = Distributed Resource Scheduler
- Automatic virtual machine placement
 - Optimize load balance across hosts
 - Decide if, when, and where to migrate
 - React to dynamic load changes
- Cluster-wide resource management
 - Scalable resource controls
 - Configurable automation levels
 - Integrated UI for all controls

DRS Can Help You...

- Manage variable loads
 - Workloads often dynamic, time-dependent
 - Quickly shift loads in response to demand
- Administer many virtual machines
 - Hierarchical organization
 - Delegated administration
- Move towards utility computing
 - Think more about aggregate resource pools
 - Think less about individual hosts

Where Does DRS Fit In?

- New product
 - Requires VirtualCenter 2 and ESX Server 3
 - Modular plug-in for VirtualCenter
- DRS module
 - Implements algorithms, enforces policies
 - Managed using VirtualCenter UI
- Leverages core technologies
 - VMotion for migrating live VMs across hosts
 - Sophisticated resource management

Key Features

- Virtual machine placement
 - Choose initial host when VM powers on
 - Dynamic rebalancing using VMotion
- Configurable automation levels
 - Manual – recommend initial host and migrations
 - Partial – automatic initial host, recommend migrations
 - Full – automatic initial host and migrations
- Resource pools
 - Flexible grouping, sharing, and isolation
 - Hierarchical organization and delegation

Talk Overview

- Context and features
- **Managing resources**
- Virtual machine placement
- System architecture
- Summary

Managing Resources

- Basic controls
 - Same as in current products
 - Shares – specify relative importance
 - Min – guaranteed resource availability
 - Max – limit resource consumption
- Resource pools
 - New feature leveraged by DRS
 - Hierarchical management

Basic Control: Shares

- Importance
 - Entitlement directly proportional to shares
 - Analogy: shares of stock in corporation
- Relative units
 - Abstract number, only ratios matter
 - Entitlement depends on total shares issued
- Named values
 - Predefine *high, normal, low* with 4 : 2 : 1 ratio
 - Defaults to *normal*

Shares Examples



- Change shares for **virtual machine**
- Dynamic reallocation



- Add **virtual machine**, overcommit resources
- Graceful degradation

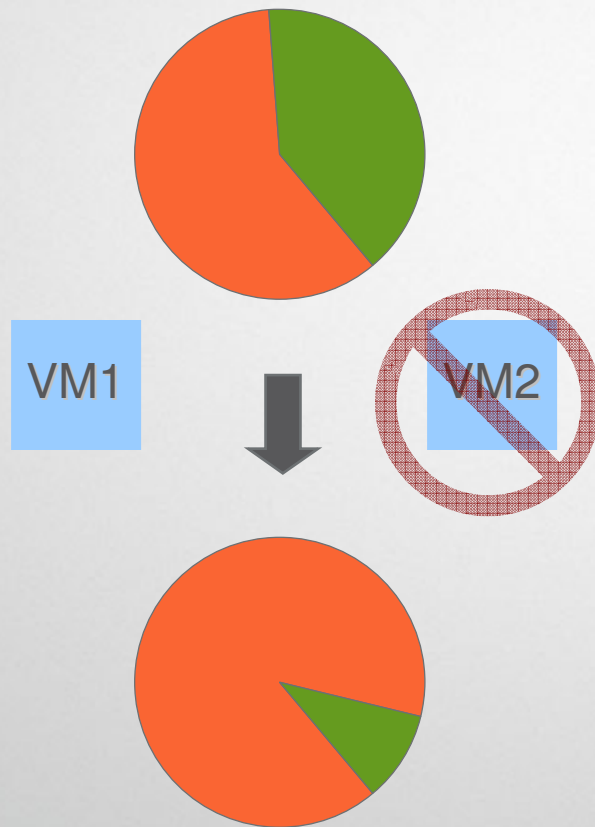


- Remove **virtual machine**
- Exploit extra resources

Basic Control: Min

- Guaranteed resources
 - Minimum service level reservation
 - Even when system overcommitted
- Absolute units
 - MHz for cpu, MB for memory
 - Defaults to zero for cpu, memory
- Virtual machine admission control
 - Reserve resources for mins
 - Sum of all VM mins \leq capacity
 - Prevent power-on if check fails

Min Example



- Total capacity
 - 600 MHz **reserved**
 - 400 MHz **available**
- Admission control
 - 2 VMs try to power-on
 - 300 MHz min each
 - Unable to admit both
- VM1 powers on
- VM2 not admitted

Basic Control: Max

- Resource limit
 - Upper bound on consumption
 - Even when system undercommitted
- Absolute units
 - MHz for CPU, MB for memory
 - Defaults to “unlimited” for cpu
 - Defaults to guest RAM size for memory

Max Example



- Current utilization
 - 600 MHz **active**
 - 400 MHz **idle**
- Start CPU-bound VM
 - 200 MHz max
 - Execution throttled
- New utilization
 - 800 MHz **active**
 - 200 MHz **idle**
 - VM prevented from using idle resources

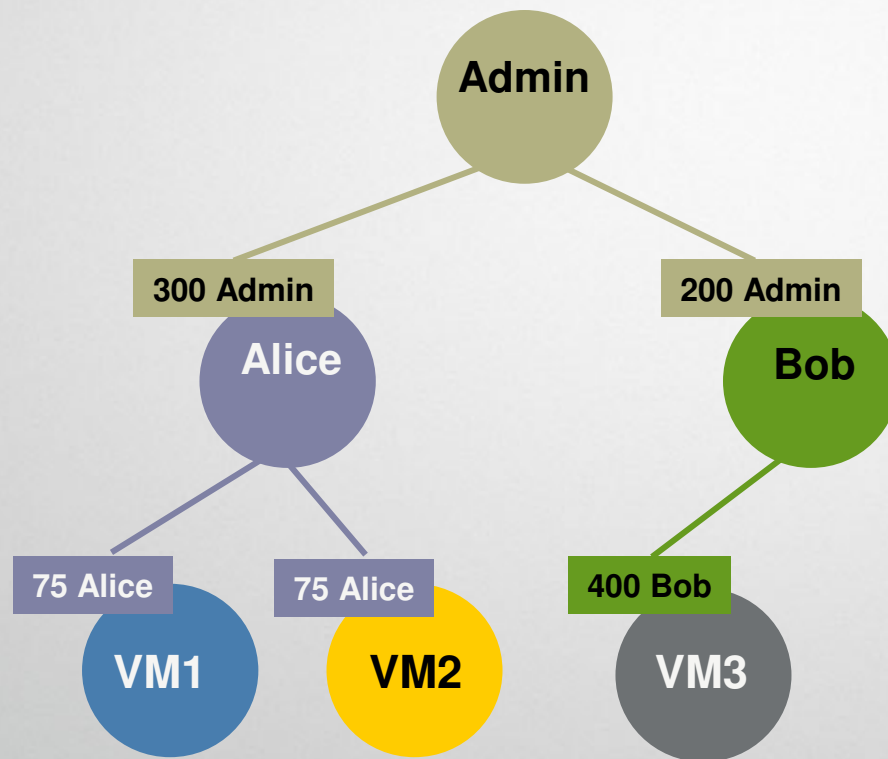
Resource Entitlements

- Resources that each VM “deserves”
 - Combining shares, min, and max
 - Allocation primarily based on shares
 - Constrained by min and max
- What if VM idles?
 - Don't give VM more than it demands
 - Resources redistributed to active VMs
 - Unused mins not wasted

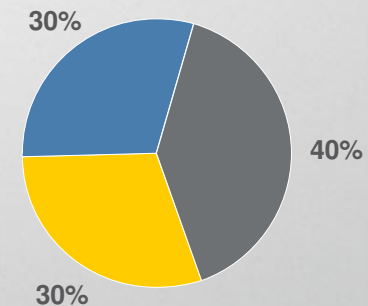
Resource Pools

- Motivation
 - Allocate aggregate resources for sets of VMs
 - Isolation between pools, sharing within pools
 - Flexible hierarchical organization
 - Access control and delegation
- What is a resource pool?
 - Named object in VirtualCenter inventory
 - Access control permissions
 - Min, max, and shares for both CPU and memory
 - Parent pool, child pools and VMs

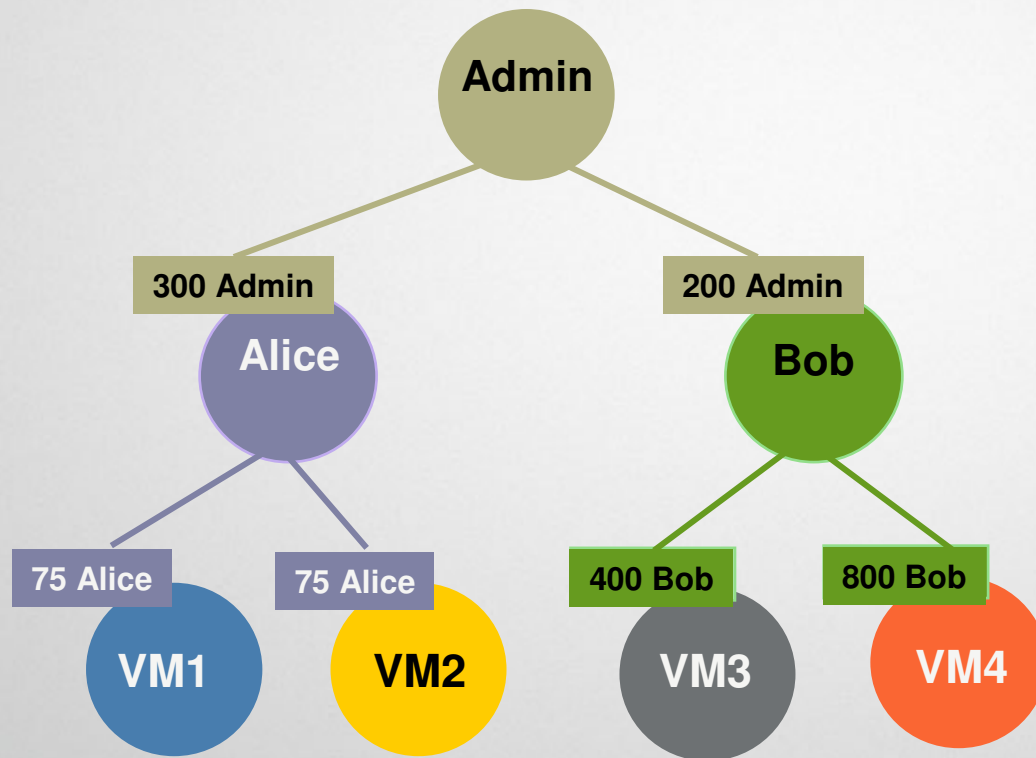
Resource Pools Example



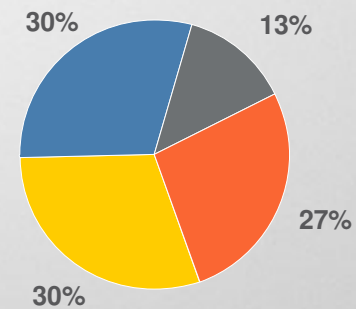
- Admin manages users
- Policy: Alice's share 50% more than Bob's
- Users manage own virtual machines
- Not shown: min, max
- VM allocations:



Example: Bob Adds Virtual Machine



- Same policy
- Pools isolate users
- Alice still gets 50% more than Bob
- VM allocations:



Resource Pool Admission Control

- Pool admission control
 - Same check as before, at pool level
 - Sum of mins for pool children \leq pool capacity
 - When create pool, power-on VM, change settings
- Growable Min option
 - Dynamically request more capacity from parent
 - Simplifies policies where hard partitions too rigid

Resource Pools UI

The screenshot displays the VMware VirtualCenter interface for the 'QA Test Resources' pool. The left pane shows the inventory tree with 'QA Test Resources' selected. The right pane shows the pool's configuration and a list of resources.

QA Test Resources Configuration:

- Guaranteed Minimum CPU: **2.6 Ghz**
- Minimum CPU Allocated: **2.3 Ghz**
- Minimum CPU Available: **0.3 Ghz**
- Total CPU Shares: **7000**
- Guaranteed Minimum Memory: **3.6 Gbytes**
- Minimum Memory Allocated: **2.2 Gbytes**
- Minimum Memory Available: **1.4 Gbytes**
- Total Memory Shares: **9000**

View Resources For: CPU Memory (selected)

Name contains: Clear

Name	Minimum (Mhz)	Maximum (Mhz)	Shares	Share Value	Grow Minimum
Beryllium & Lithium	1200		High	4000	Yes
Virtual Center 2	450		Low	1000	No
Test Case Resources	450	750	Low	1000	No
GSX 4 Testing	200		Low	1000	No
Redhat Linux 9.0	1500		High	4000	
Windows 2003 Server Std Edition	750		Medium	2000	
Redhat Linux 9.0	667		Low	1000	
Windows 2003 Server Std Edition	233		Custom	2400	
Linux 04-19-1	1200		High	4000	
ravi-tmp-test-1-kguzik-esx.vmware.com	450		Low	1000	
Windows 2003 Server Std Edition	450	750	Low	1000	
ravi-tmp-test-4-kguzik-esx.vmware.com	200		Low	1000	
Redhat Linux 9.0	1500		High	4000	
Linux7	750		Medium	2000	
Linux VM created by vpx:11	667		Low	1000	
Linux3 04-19					

△ Show system status Connected as kguzik

Delegated Administration

- Cluster administrator
 - Default pool contains all cluster resources
 - Aggregate cpu and memory capacity of all hosts
 - Carves up cluster resources into pools
 - Provides bulk allocations to pool administrators
- Pool administrator
 - Pool may reflect department, project, client, etc.
 - Carves up pool resources into smaller pools for users
- End user
 - Allocates resources from personal pool to virtual machines
 - View restricted to personal pool hierarchy

Best Practices

- Use Mins and Shares appropriately
 - Shares generally more flexible policy tool
 - Use shares to isolate without hard partitioning
 - Use mins to guarantee acceptable service
- Maintain some spare capacity
 - Don't use mins that commit entire cluster
 - Slack for maintenance, rebalancing
 - Needed to tolerate host failures

Talk Overview

- Context and features
- Managing resources
- **Virtual machine placement**
- System architecture
- Summary

Virtual Machine Placement

- Goals
 - Balance virtual machine load across hosts in cluster
 - Enforce resource policies accurately
 - Respect placement constraints
- Dynamic balancing
 - Monitor key virtual machine, pool, and host metrics
 - Deliver entitled resources to pools and VMs
 - Recommend migrations (prioritized list)
- Initial placement
 - Power on virtual machine in resource pool
 - Recommend host (prioritized list)

Placement Constraints

- VMotion compatibility
 - Processor type
 - SAN and LAN connectivity
- Anti-affinity rules
 - Run virtual machines on different hosts
 - Motivation: high-availability, clustering
- Affinity rules
 - Run virtual machines on same host
 - Motivation: locality, performance benefits

Dynamic Balancing

- What to balance?
 - Load, adjusted for resource entitlement
 - Load = utilization, if all VMs equally important
- When to balance?
 - Re-evaluate every few minutes
 - Changes to pool or VM settings
 - Add or remove host
- Aggressiveness
 - Migration rate, recommendation strength
 - Depends on severity of imbalance

Balancing Details

- Compute virtual machine entitlements
 - Based on pool and virtual machine resource allocations
 - Don't give virtual machine more than it demands
 - Reallocate extra resources fairly
- Compute host loads
 - Sum entitlements for virtual machines on host
 - Normalize by host capacity
- Consider possible VMotions
 - Evaluate effect on cluster balance
 - Evaluate migration cost for involved hosts
- Recommend best moves (if any)

Dynamic Balancing UI

A Cluster

Summary
Virtual Machines
Hosts
Relationships
Performance
Tasks & Events
Alarms
Permissions

General

Dynamic Resource Scheduling: Enabled
Distributed Availability Services: Enabled

Number of Hosts: 12
Total Processors: 24
Total CPU: 36 GHz
Total Memory: 24 GB

Number of Virtual Machines: 37
Running Virtual Machines: 33
Total Migrations: 102
Active Migrations: 3

Dynamic Resource Scheduling (DRS)

Automation Level: Partially Automated
Migration Rate: Moderate

Commands

[Edit Properties](#)
[Refresh](#)

Distributed Availability Services (DAS)

Admission Control: Allow constraint violations
Configured Failover Capacity: 3 Hosts
Current Failover Capacity: 3 Hosts (4.6 GHz 2.8 MB)

DRS Resource Distribution

DRS Migration Recommendations

Priority	Virtual Machine	Reason	Current Host	Target Host	CPU Load		Memory Load	
★★★★	Solaris 9 (experime...	Improve CPU fairness	esx001.vm...	esx066.v...	<div style="width: 84.1%; background-color: #0070c0; height: 10px;"></div> 84.1	<div style="width: 54.3%; background-color: #e69d00; height: 10px;"></div> 54.3	<div style="width: 97.3%; background-color: #0070c0; height: 10px;"></div> 97.3	<div style="width: 34.3%; background-color: #e69d00; height: 10px;"></div> 34.3
★★★	MS-DOS	Improve CPU fairness	esx012.vm...	esx041.v...	<div style="width: 84.1%; background-color: #0070c0; height: 10px;"></div> 84.1	<div style="width: 54.3%; background-color: #e69d00; height: 10px;"></div> 54.3	<div style="width: 97.3%; background-color: #0070c0; height: 10px;"></div> 97.3	<div style="width: 34.3%; background-color: #e69d00; height: 10px;"></div> 34.3
★	MS-DOS 3	Improve Memory fair...	esx009.vm...	esx072.v...	<div style="width: 84.1%; background-color: #0070c0; height: 10px;"></div> 84.1	<div style="width: 54.3%; background-color: #e69d00; height: 10px;"></div> 54.3	<div style="width: 97.3%; background-color: #0070c0; height: 10px;"></div> 97.3	<div style="width: 34.3%; background-color: #e69d00; height: 10px;"></div> 34.3

Apply Recommendation Show Performance Charts

■ Virtual machine
 ■ Current host total
 ■ Target host total

Migration History

Past Hour

Initial Placement UI

The screenshot shows the VMware VirtualCenter interface. The main window is titled 'localhost - VMware VirtualCenter'. The 'Inventory' pane on the left shows a tree view of 'Hosts and Clusters'. The 'Staging and Test Cluster' is selected, showing its sub-entities: Hosts, Staging and Test Cluster Resources, Staging Resources, Development Pre-Test Resources, Engineering Test Resources, QA Test Resources, Beryllium & Lithium, VirtualCenter 2.0, Test Case Resources, and GSX 4 Testing. The 'Advanced Projects Cluster', 'Engineering and QA Cluster', and 'Standalone Hosts' are also visible.

The 'Staging and Test Cluster' window is open, showing a search bar and a list of hosts. A 'Virtual Machine Host Selection' dialog box is overlaid on top, prompting the user to choose a host. The dialog contains the following table:

Hosts	Recommendation	Ave CPU	Ave Memory
esx010.eng.vmware.com	★★★★★	112 Mhz	1450
FS2.vmware.com	★★★★★	385 Mhz	1011
KEN-GUZIK-2K.vmware.com	★★★	639 Mhz	454
esx009.eng.vmware.com	★★	1265 Mhz	123
exit27.vmware.com	★	5610 Mhz	8469

The dialog also includes 'OK' and 'Cancel' buttons. The background window shows a list of hosts with their status (e.g., Powered on, Powered off) and a 'Status' column with colored indicators.

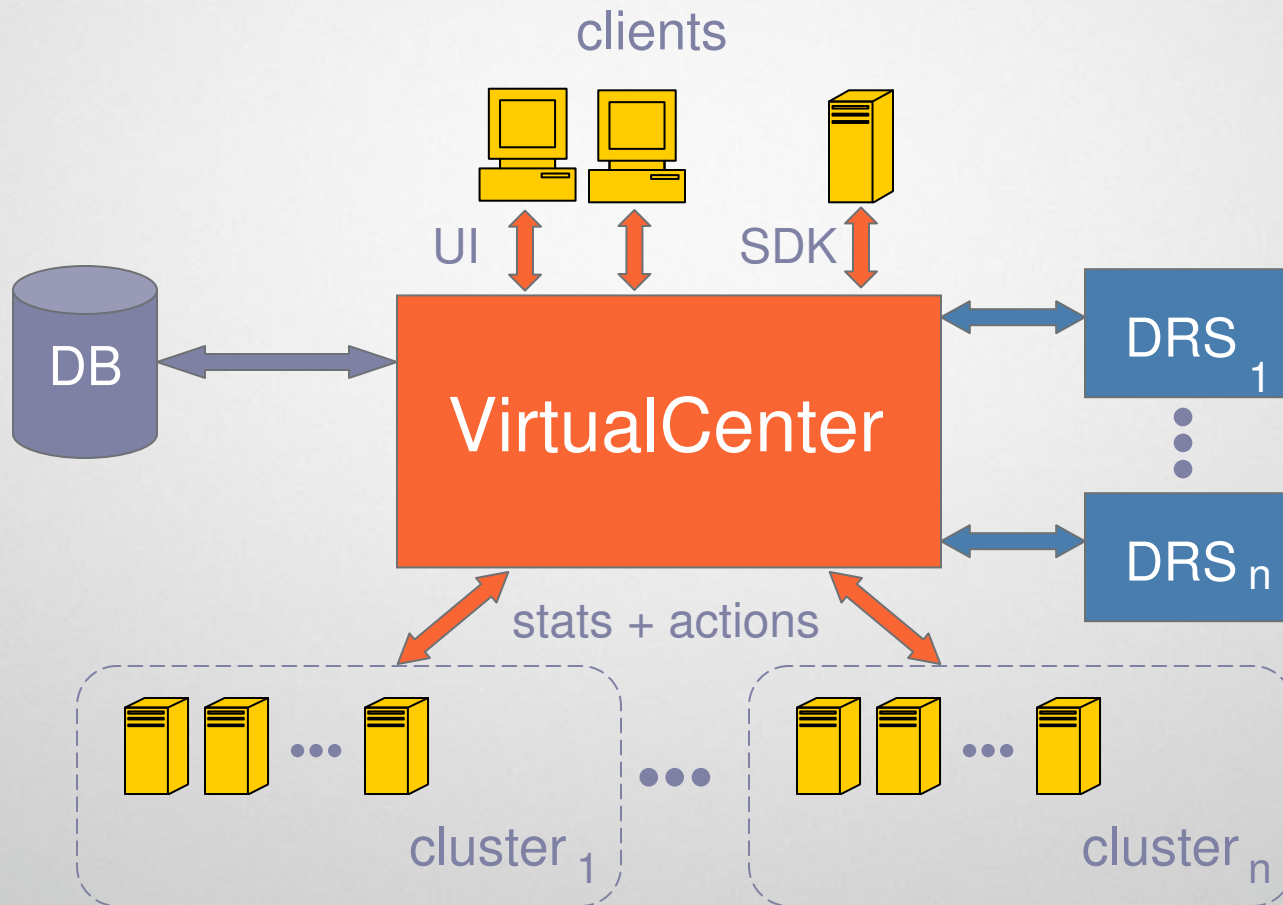
Best Practices

- Follow strong recommendations
 - Otherwise balance and fairness may deteriorate
 - Some VMotion is necessary
- Enable automation
 - Choose default based on environment, comfort level
 - Use per-VM automation level overrides
 - Let DRS autonomously manage most VMs
 - Can keep human in loop for critical VMs

Talk Overview

- Context and features
- Managing resources
- Virtual machine placement
- **System architecture**
- Summary

System Architecture Overview



System Architecture Constraints

- Cluster size
 - LAN, not WAN
 - Up to 32 hosts per cluster
 - Host capacities may differ significantly
- Time scale
 - Minutes, not milliseconds
 - VMotion VM downtime \approx milliseconds, but end-to-end latency \approx tens of seconds
 - Migrate VM infrequently \approx minutes to hours
- Algorithm performance
 - Milliseconds, not minutes
 - Operations occur at human time scale

Summary

- Automatic virtual machine placement
 - Recommendations or full automation
 - Initial placement at virtual machine power-on
 - Dynamic load balancing
- Powerful resource controls
 - Flexible cluster-wide policies
 - Hierarchical resource pools
 - Virtual machine affinity rules
- Future directions
 - Integrated I/O bandwidth management
 - Detect longer-term trends, proactive migration