VMworld 2005

virtualize now

las vegas • october 18-20, 2005

VMworld2005
virtualizenow

PAC485
Managing Datacenter Resources Using the VirtualCenter Distributed Resource Scheduler

Carl Waldspurger
Principal Engineer, R&D

VMworld2005
virtualizenow

This presentation may contain VMware confidential information.

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

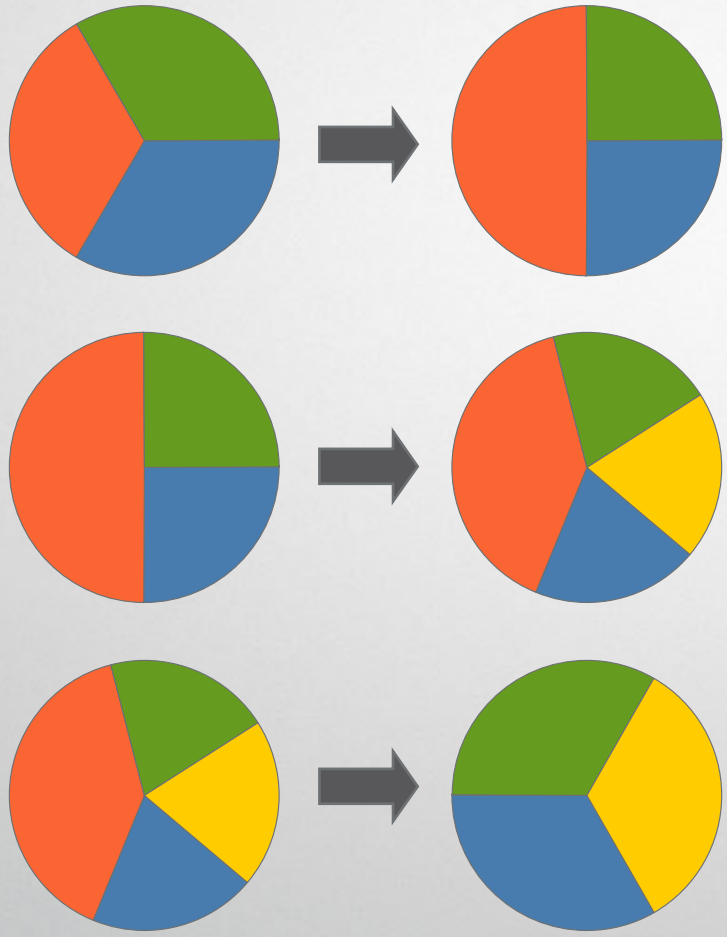
Managing Resources

- 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

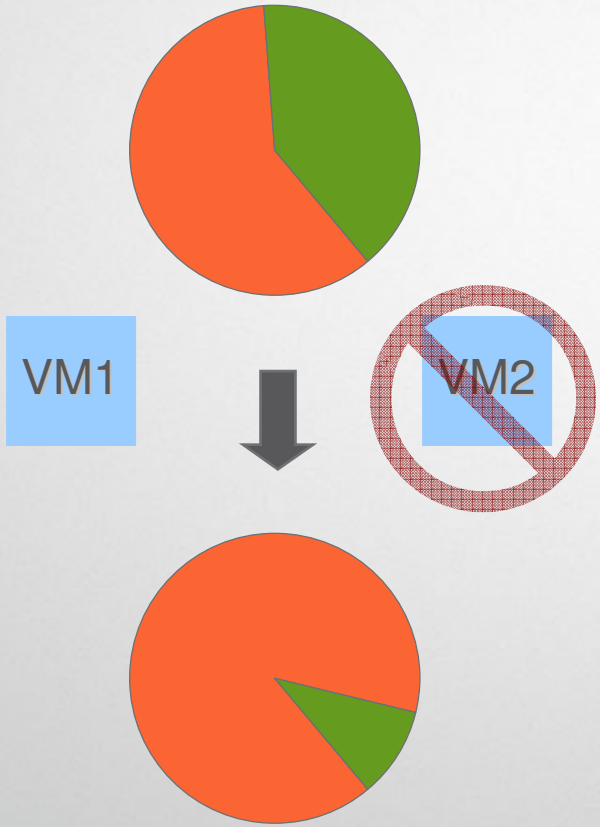
- Add **virtual machine**, overcommit resources
- Graceful degradation

- 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 ≤ capacity
  - Prevent power-on if check fails

Min Example

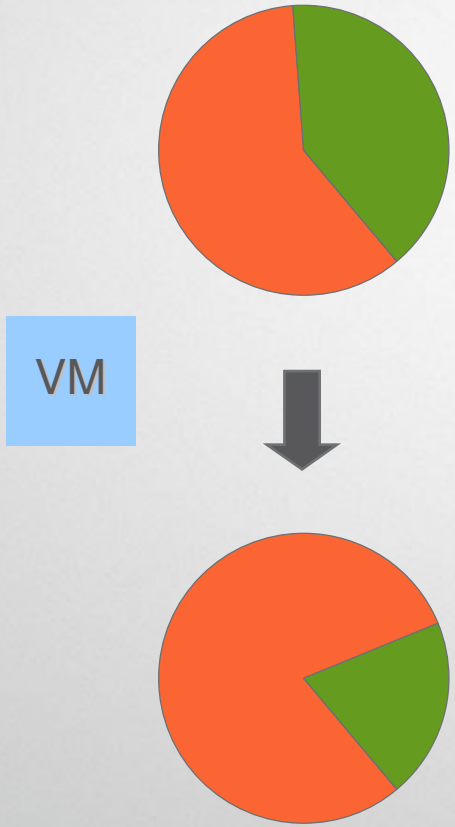


- Total capacity
  - 600 MHz **reserved**
  - 400 MHz **available**
- Admission control
  - 2 VMs try to power-on
  - 300 MHz min each
  - Unable to admit both
- VM1 powers on
- VM2 not admitted

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

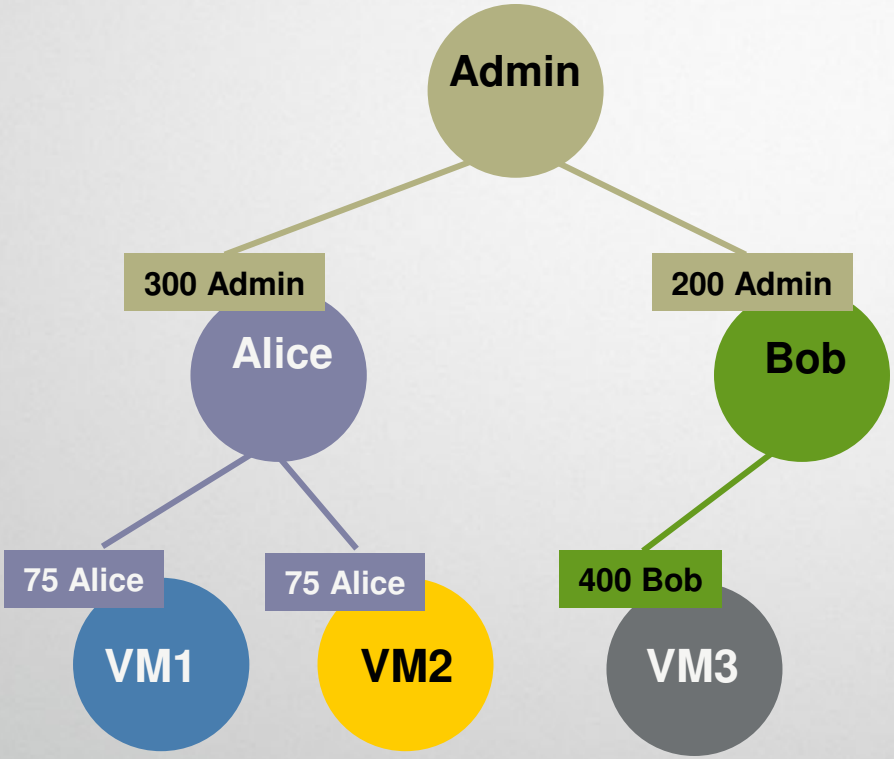
Resource Entitlements

- 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

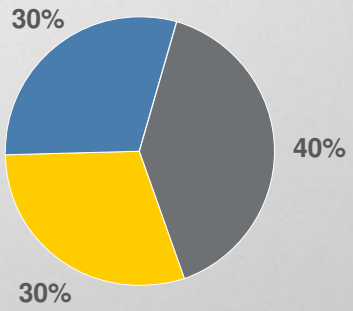
Resource Pools

- 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

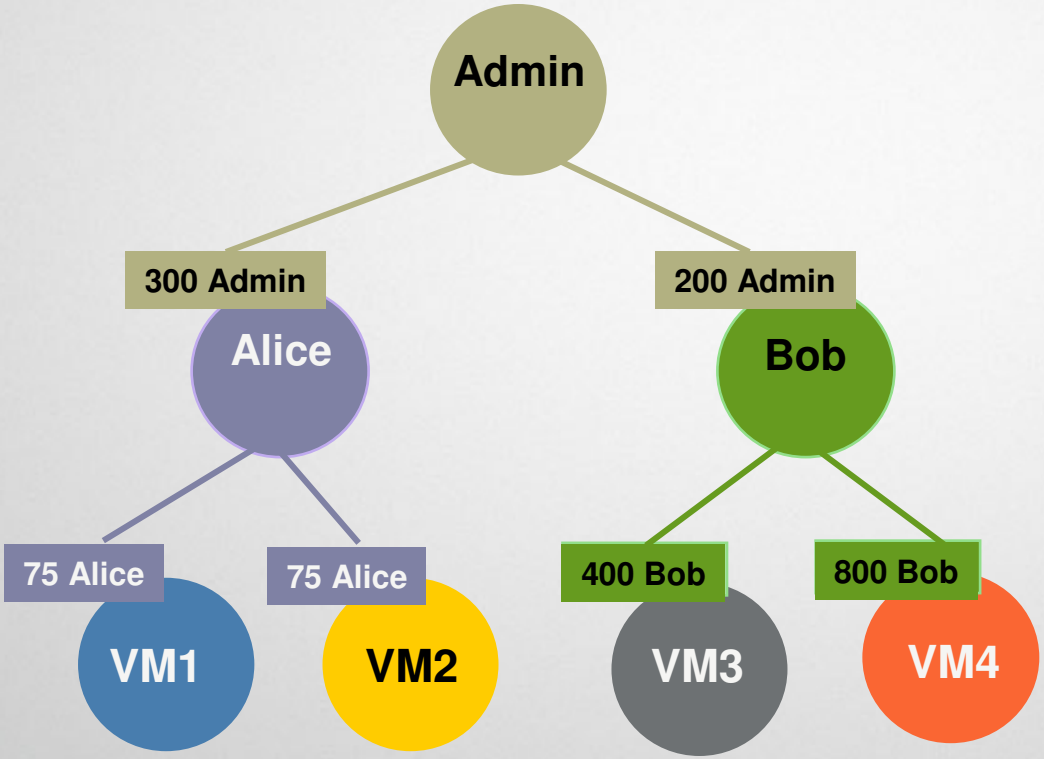
Resource Pools Example



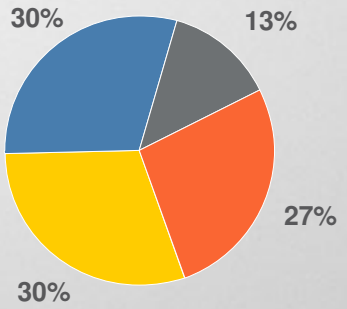
- Admin manages users
- Policy: Alice's share 50% more than Bob's
- Users manage own virtual machines
- Not shown: min, max
- VM allocations:



Example: Bob Adds Virtual Machine



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

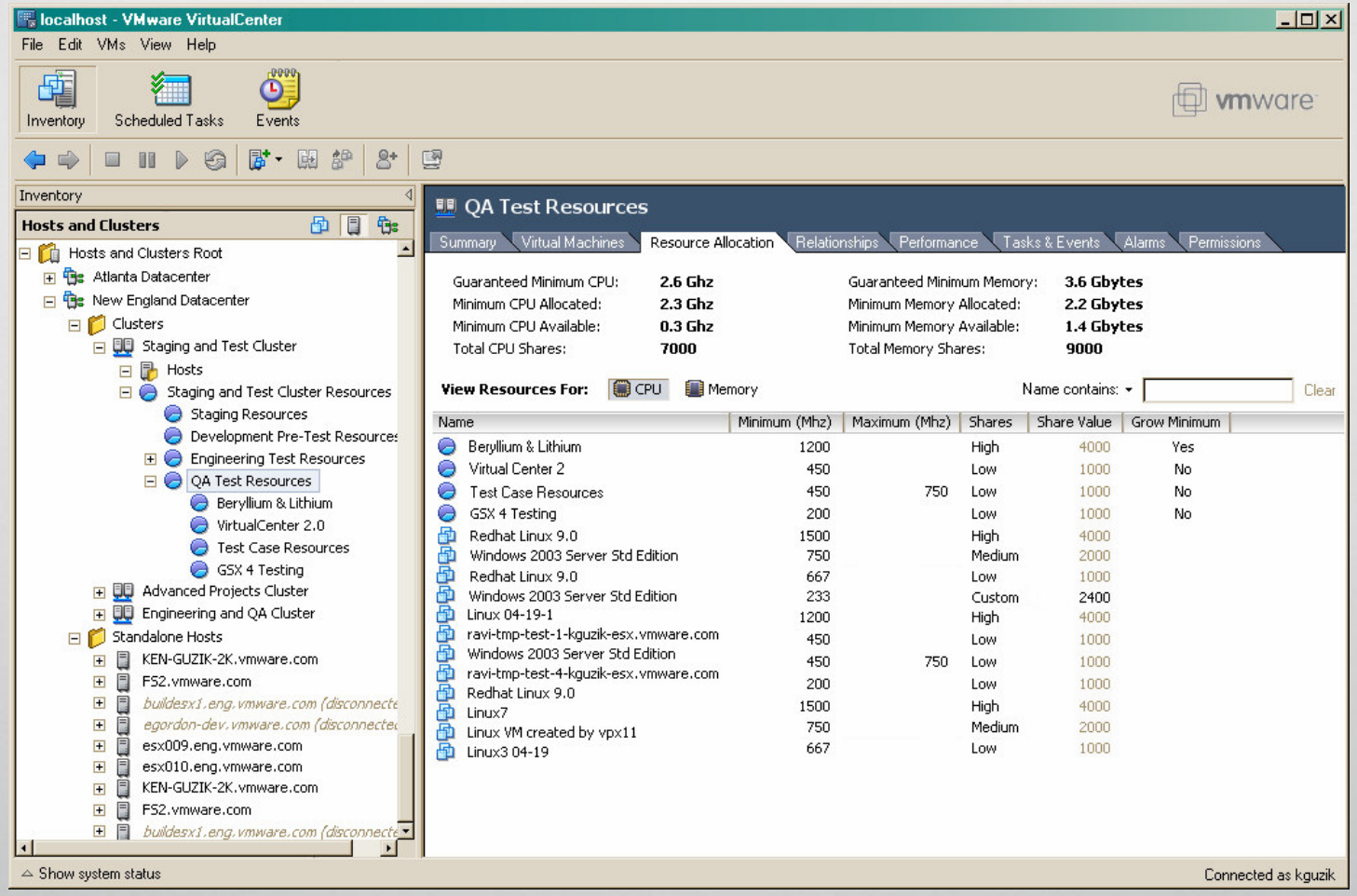


Resource Pool Admission Control

- Pool admission control
  - Same check as before, at pool level
  - Sum of mins for pool children ≤ 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

VMworld2005 virtualizenow

Resource Pools UI



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

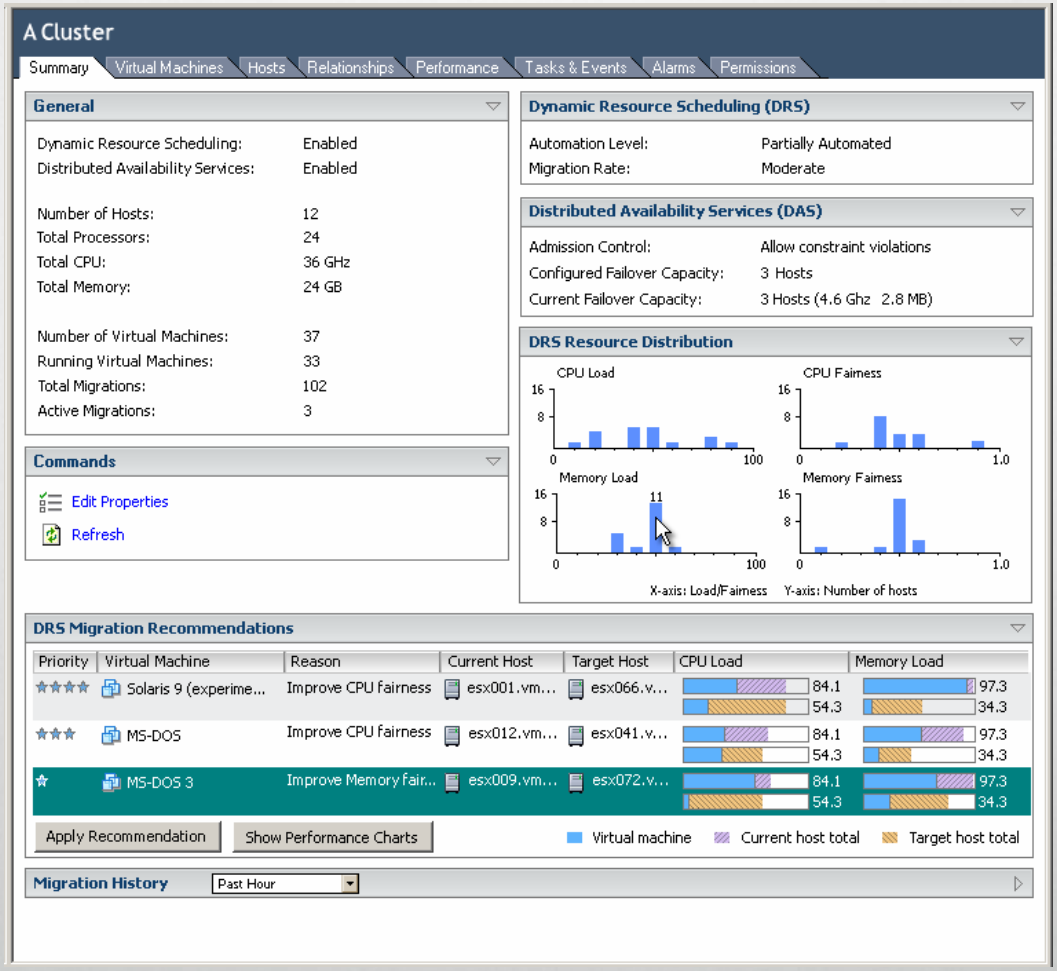
- What to balance?
  - Load, adjusted for resource entitlement
  - Load = utilization, if all VMs equally important
- When to balance?
  - Re-evaluate every few minutes
  - Changes to pool or VM settings
  - Add or remove host
- Aggressiveness
  - Migration rate, recommendation strength
  - Depends on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

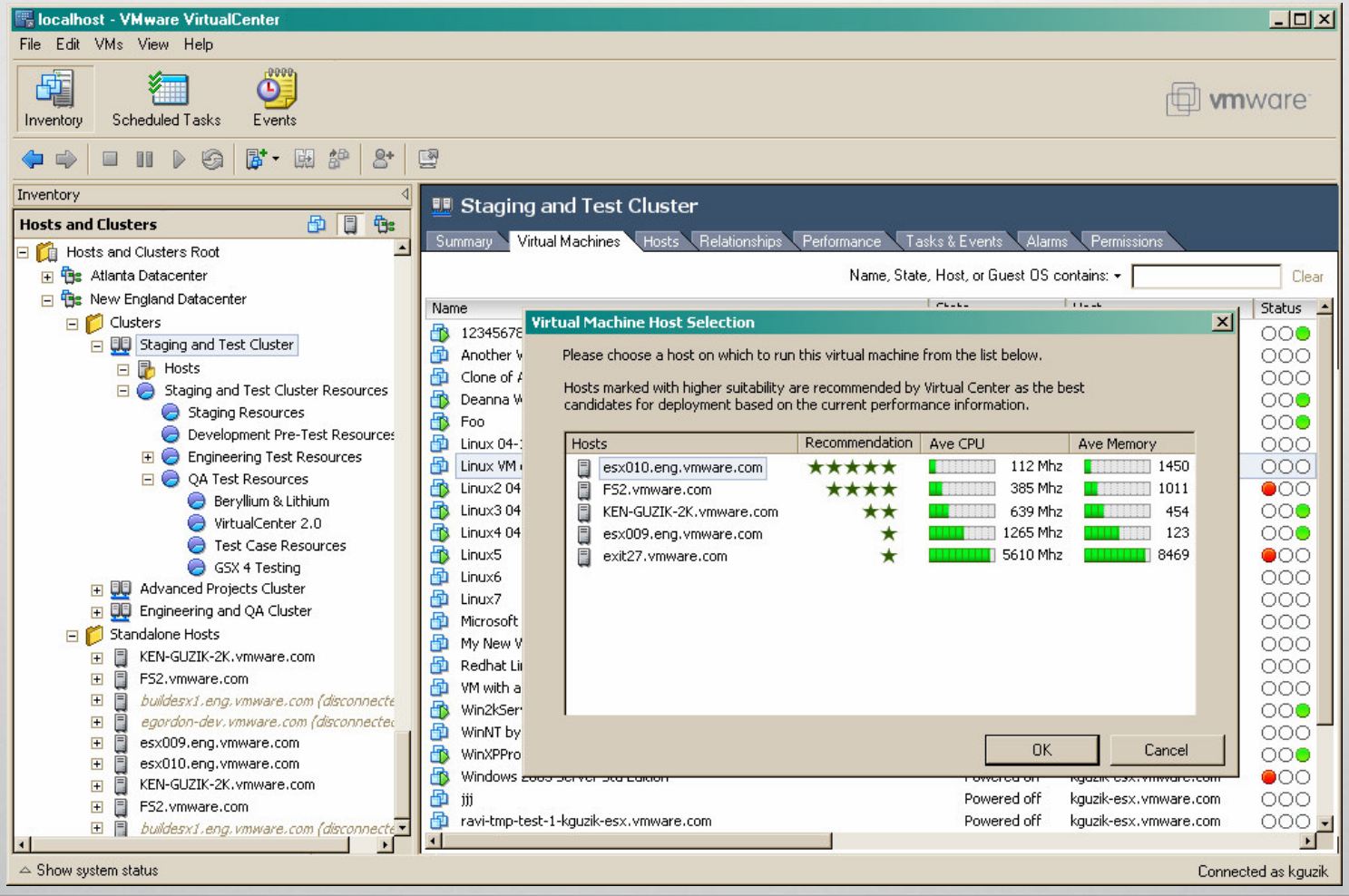
VMworld2005 virtualizenow

Dynamic Balancing UI



VMworld2005 virtualizenow

Initial Placement UI



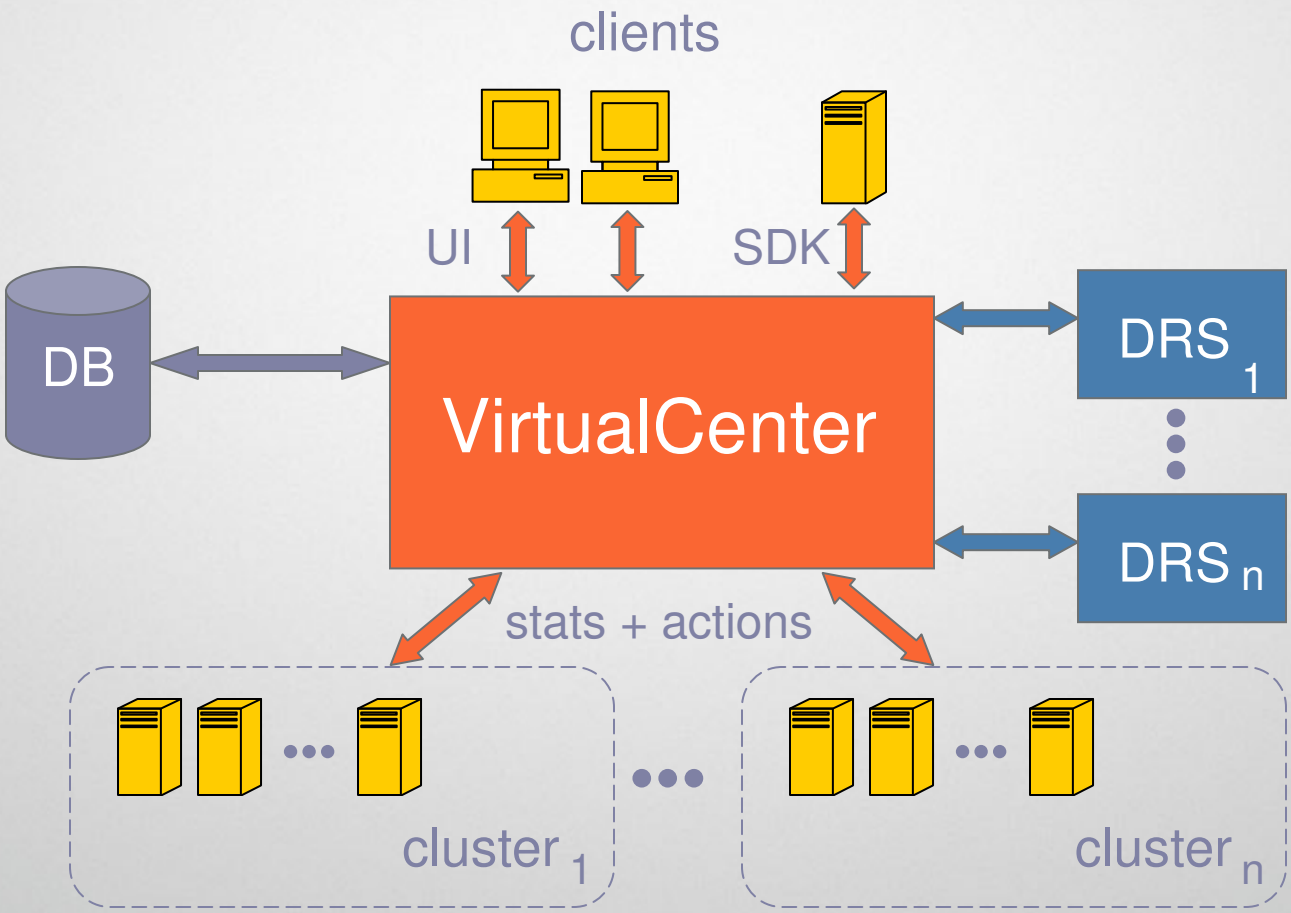
Best Practices

- 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 ≈ milliseconds,
    but end-to-end latency ≈ tens of seconds
  - Migrate VM infrequently ≈ 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