Solving (NP-Hard) Scheduling Problems with oVirt & OptaPlanner

Jason Brooks
Red Hat Open Source & Standards
SCALE13x, Feb 2015

What Is oVirt?

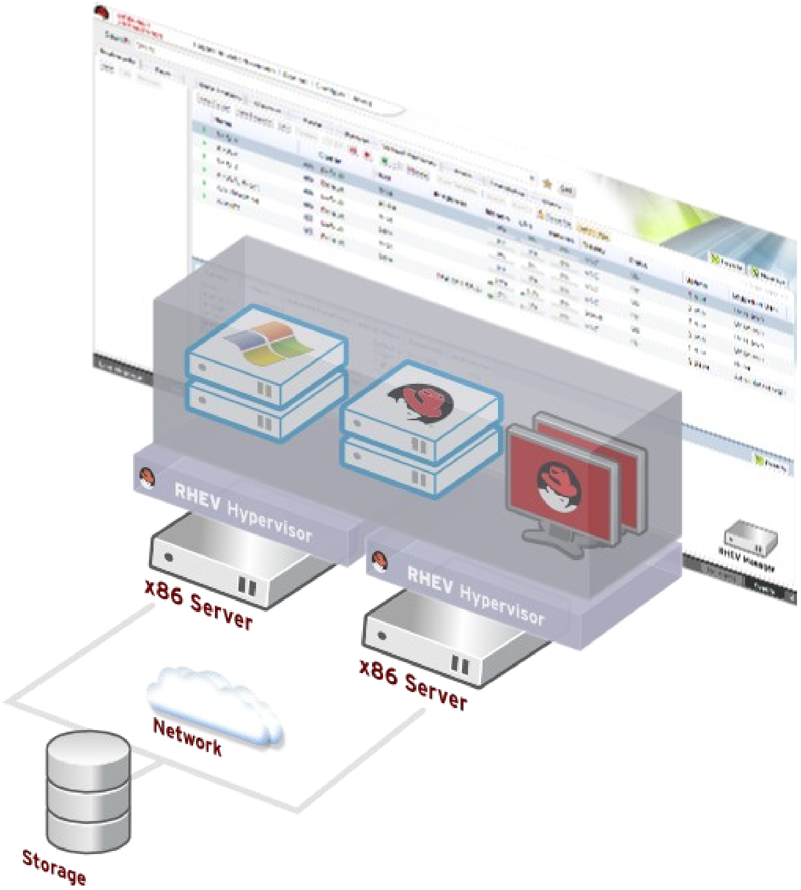
Large scale, centralized management for server and desktop virtualization

Based on leading performance, scalability and security infrastructure technologies

Provide an open source alternative to vCenter/vSphere

Focus on KVM for best integration/performance

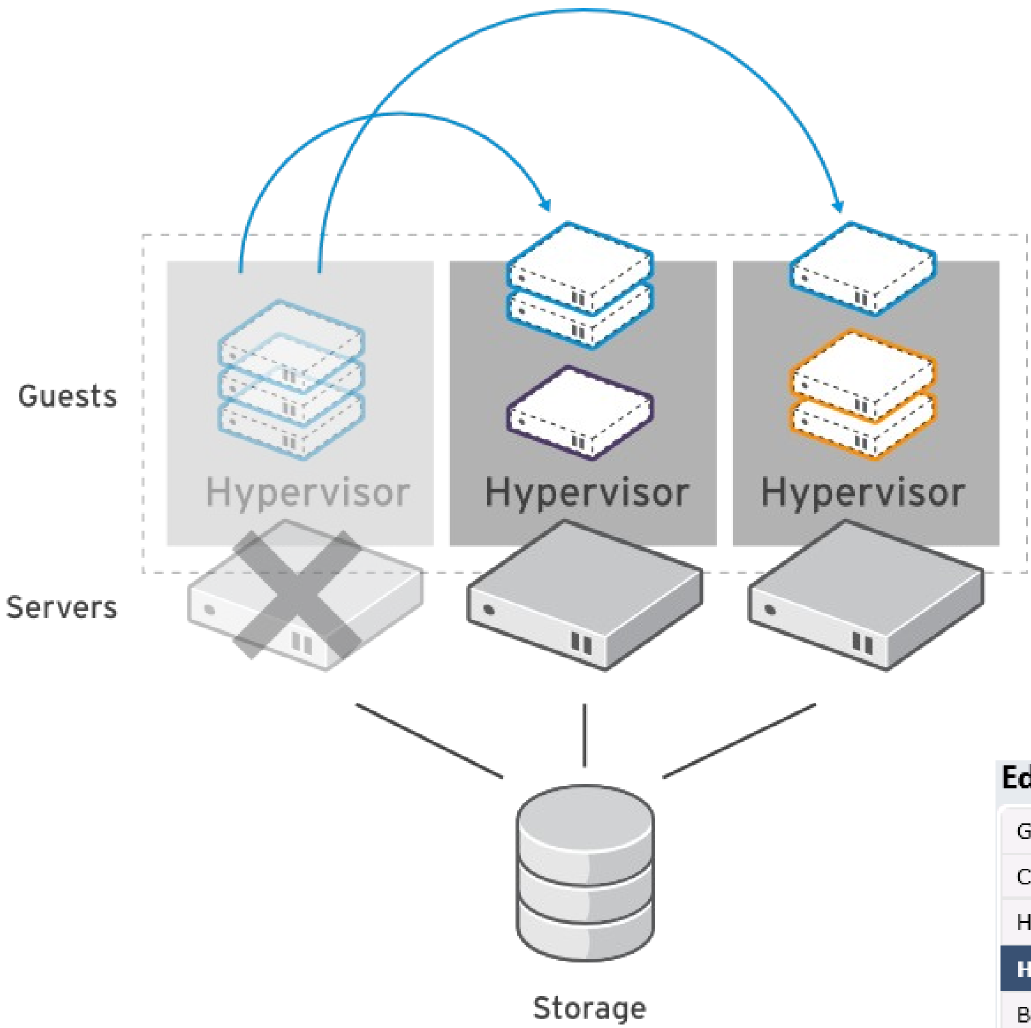
Focus on ease of use/deployment



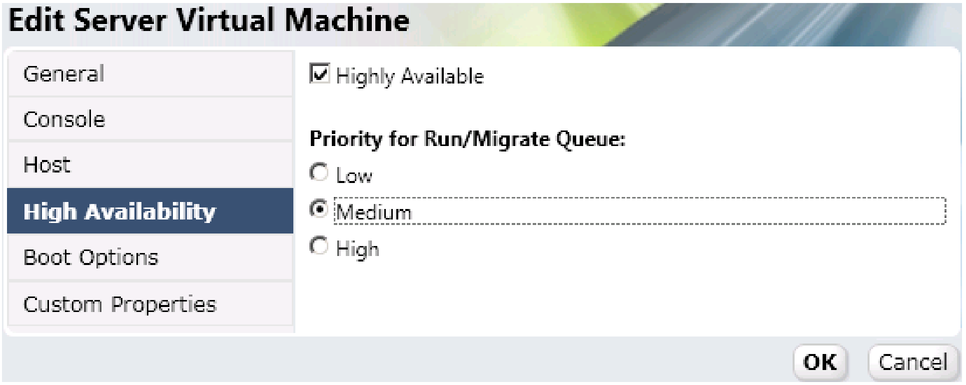
Virt & Cloud Scheduling

- Running a new VM
- Selecting migration destination
- Load balancing

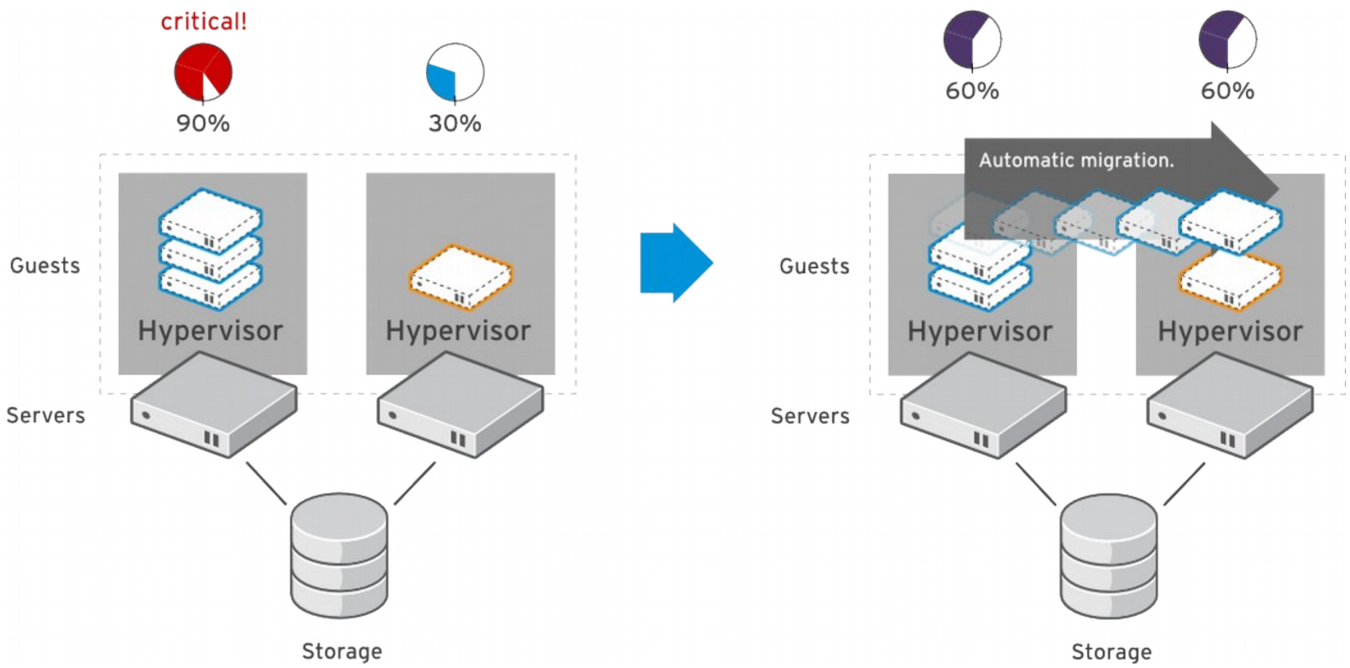
High Availability



- Build a highly available enterprise infrastructure
- Continually monitor host systems and virtual machines
- Automatically restart virtual machines in case of host failure
  - Restart virtual machine on another node in the cluster
- Use live migration to “fail-back” a VM to it's original host when the server is restored



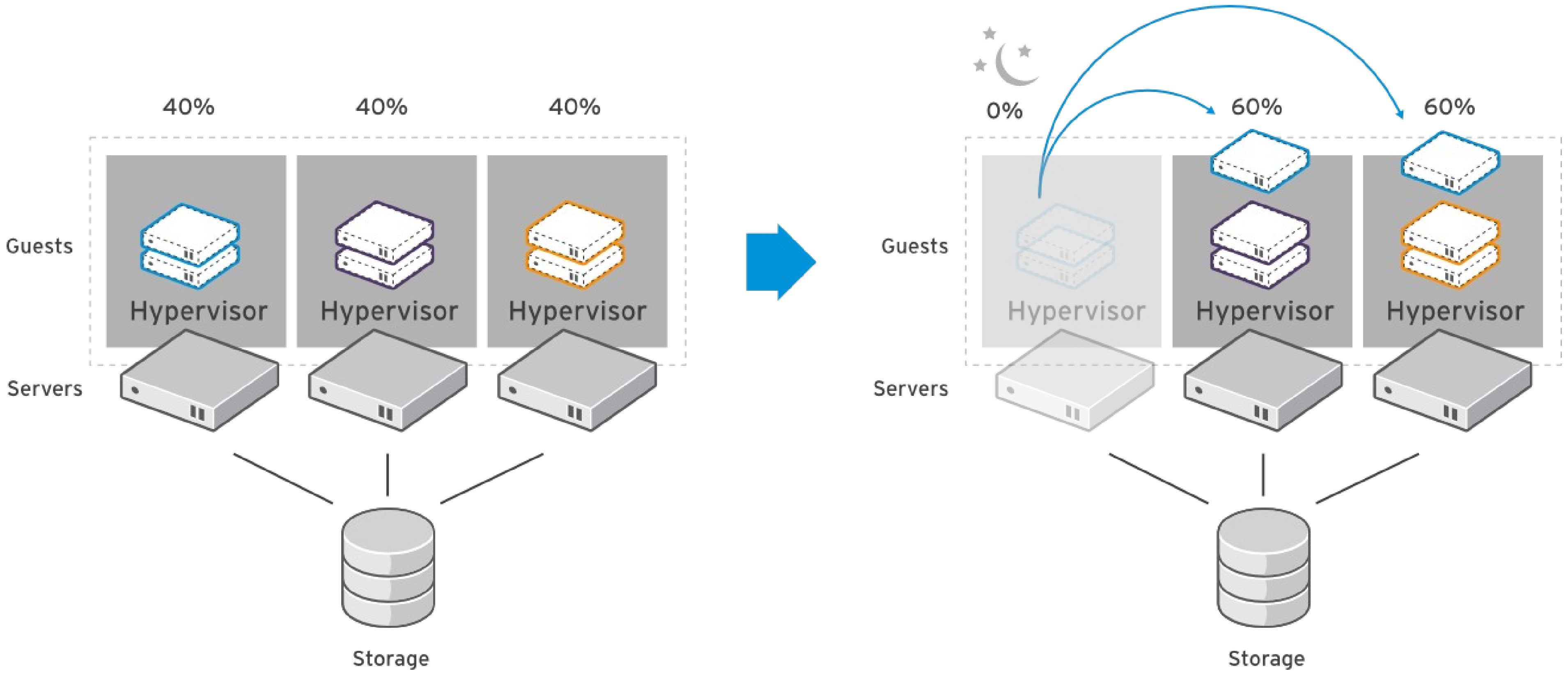
System Scheduler



- Dynamically balance workloads in the data center.
- Automatically live migrate virtual machines based on resources
- Define custom policies for distribution of virtual machines

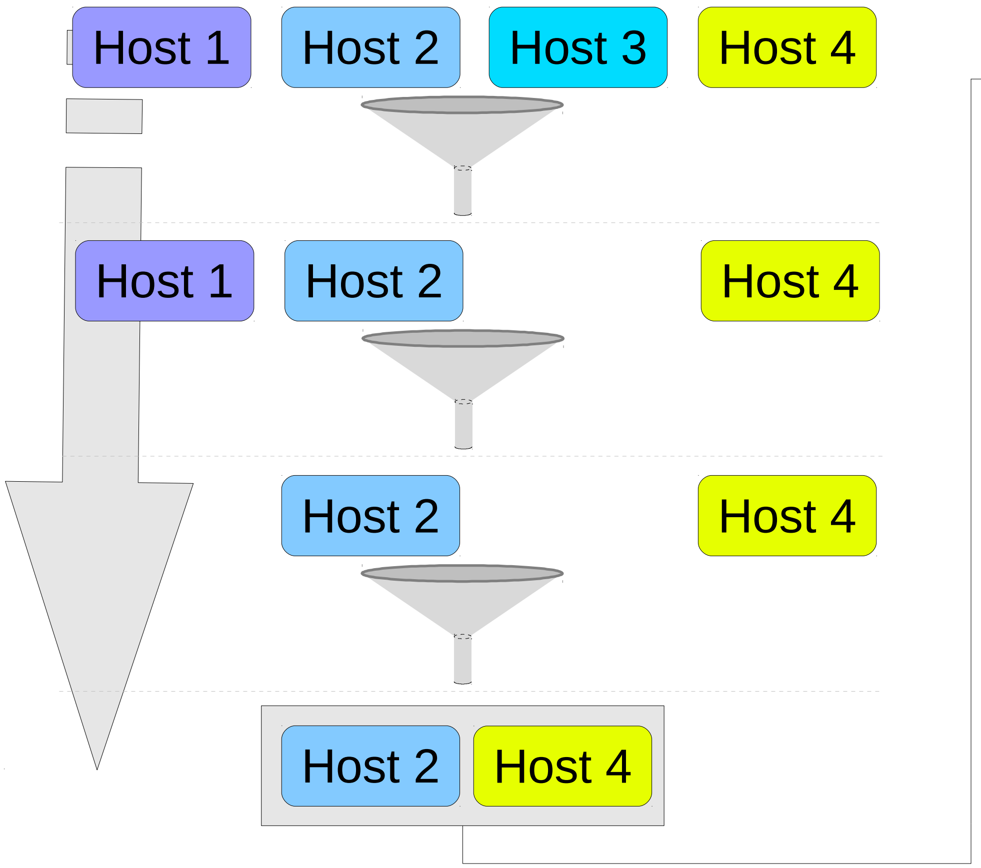
Maintain consistent resource usage across the enterprise data center

Power Saver



Define policies to optimize workload on a fewer number of servers during “off-peak” hours

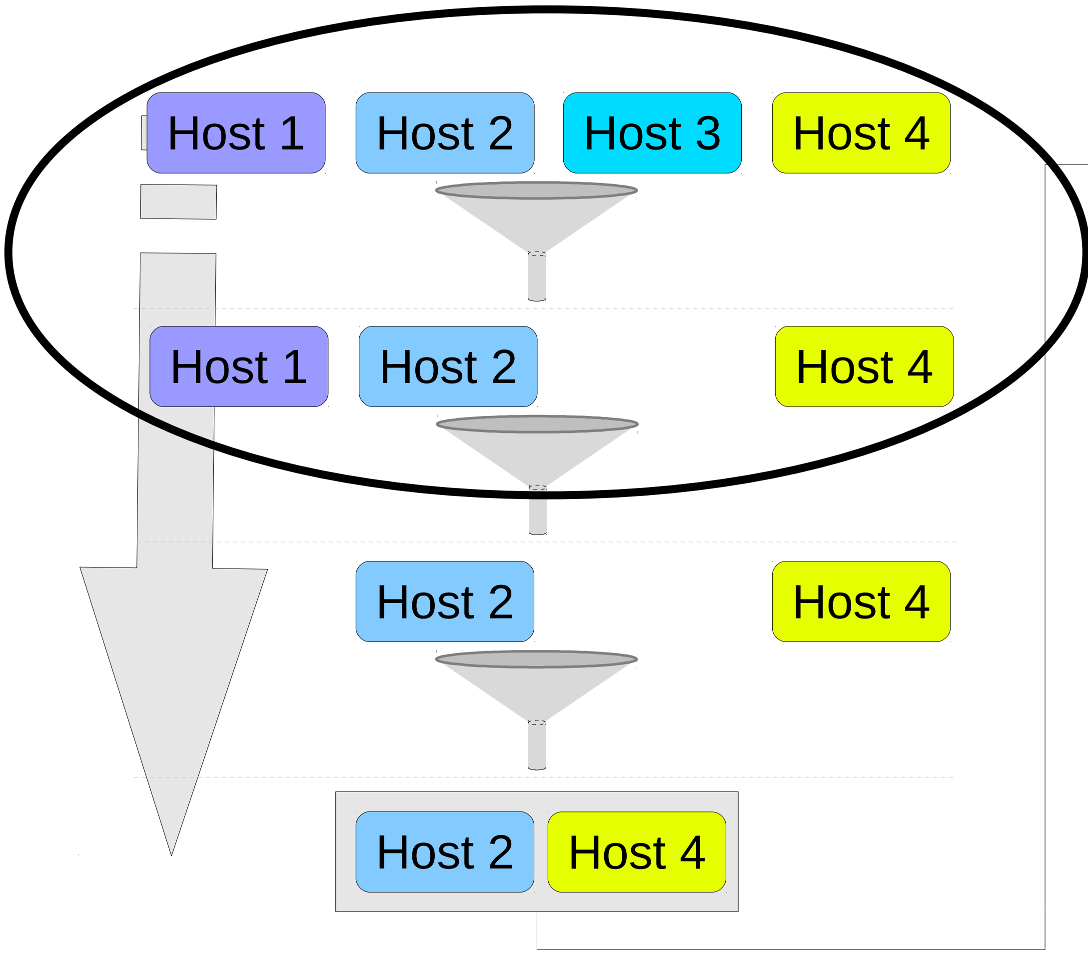
Scheduling in oVirt



	func 1	func 2	sum
Factor	5	2	
Host 2	10	2	54
Host 4	3	12	**39***

***Host 4 sum: 3*5+12*2 = 39**

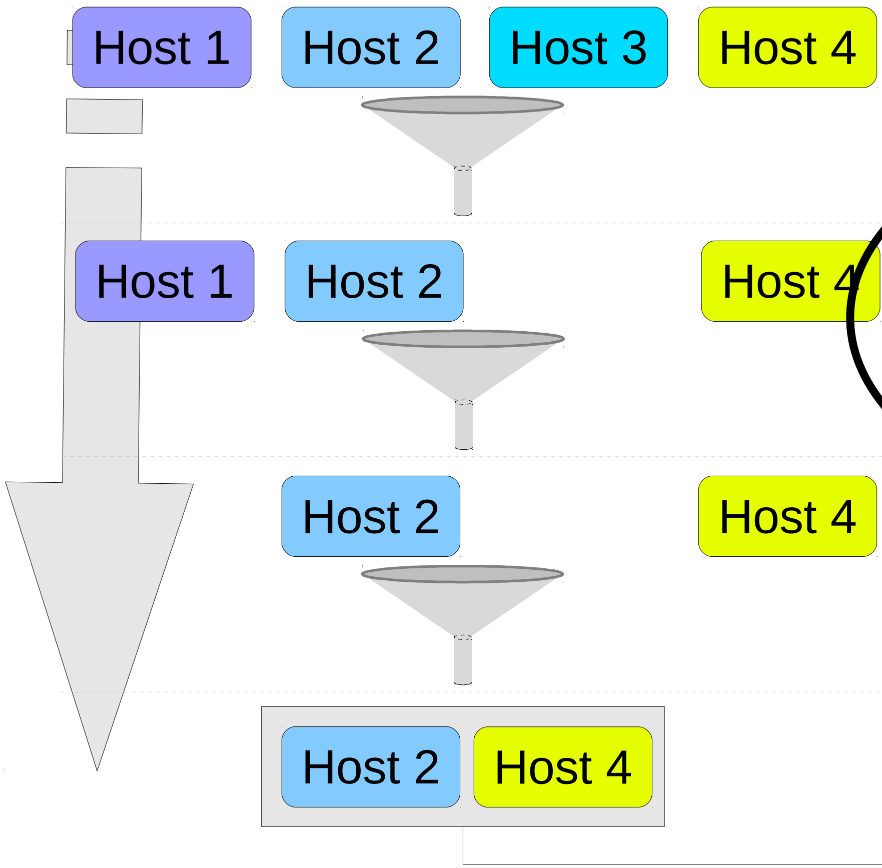
Filters



	func 1	func 2	sum
Factor	5	2	
Host 2	10	2	54
Host 4	3	12	**39***

***Host 4 sum: 3*5+12*2 = 39**

Weights

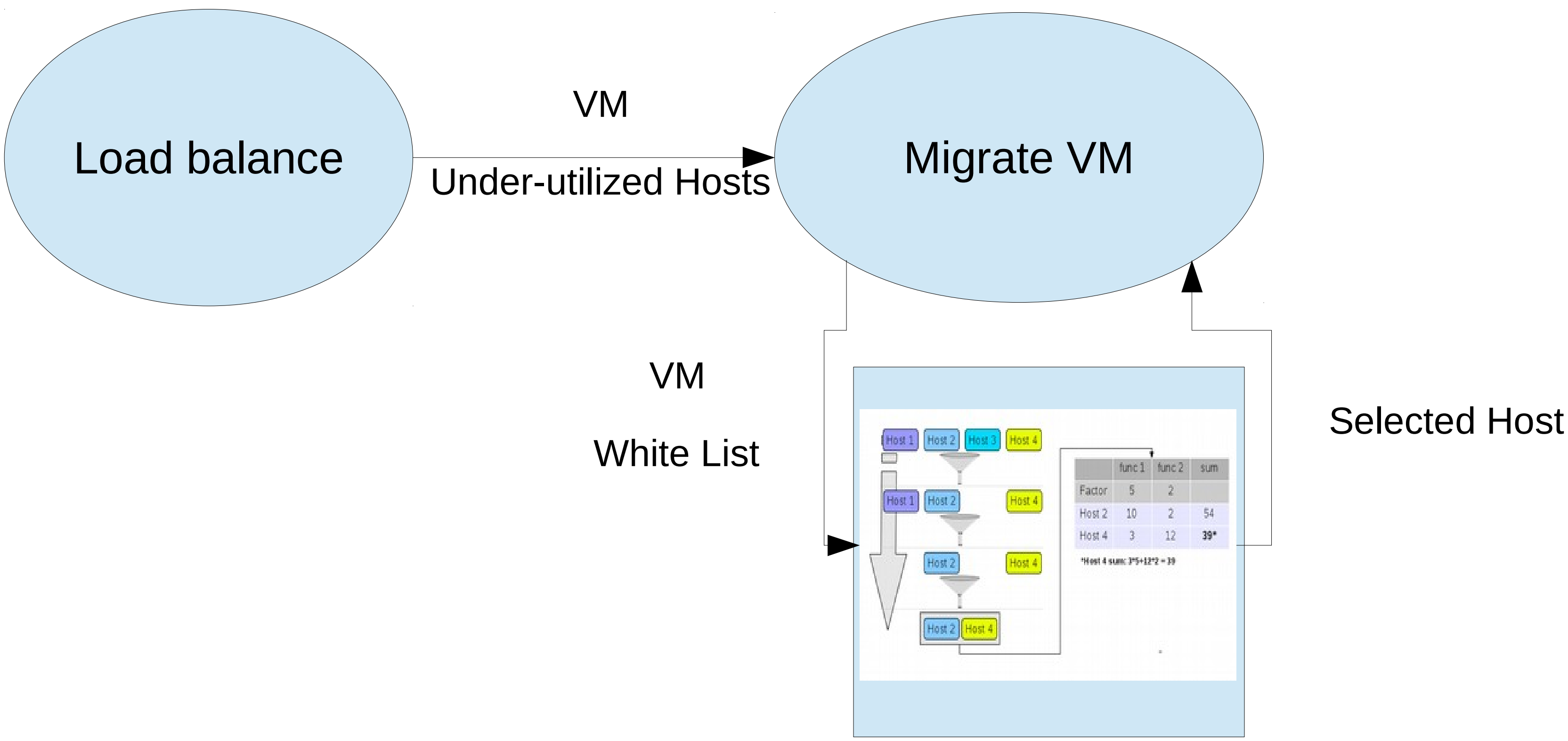


	func 1	func 2	sum
Factor	5	2	
Host 2	10	2	54
Host 4	3	12	**39***

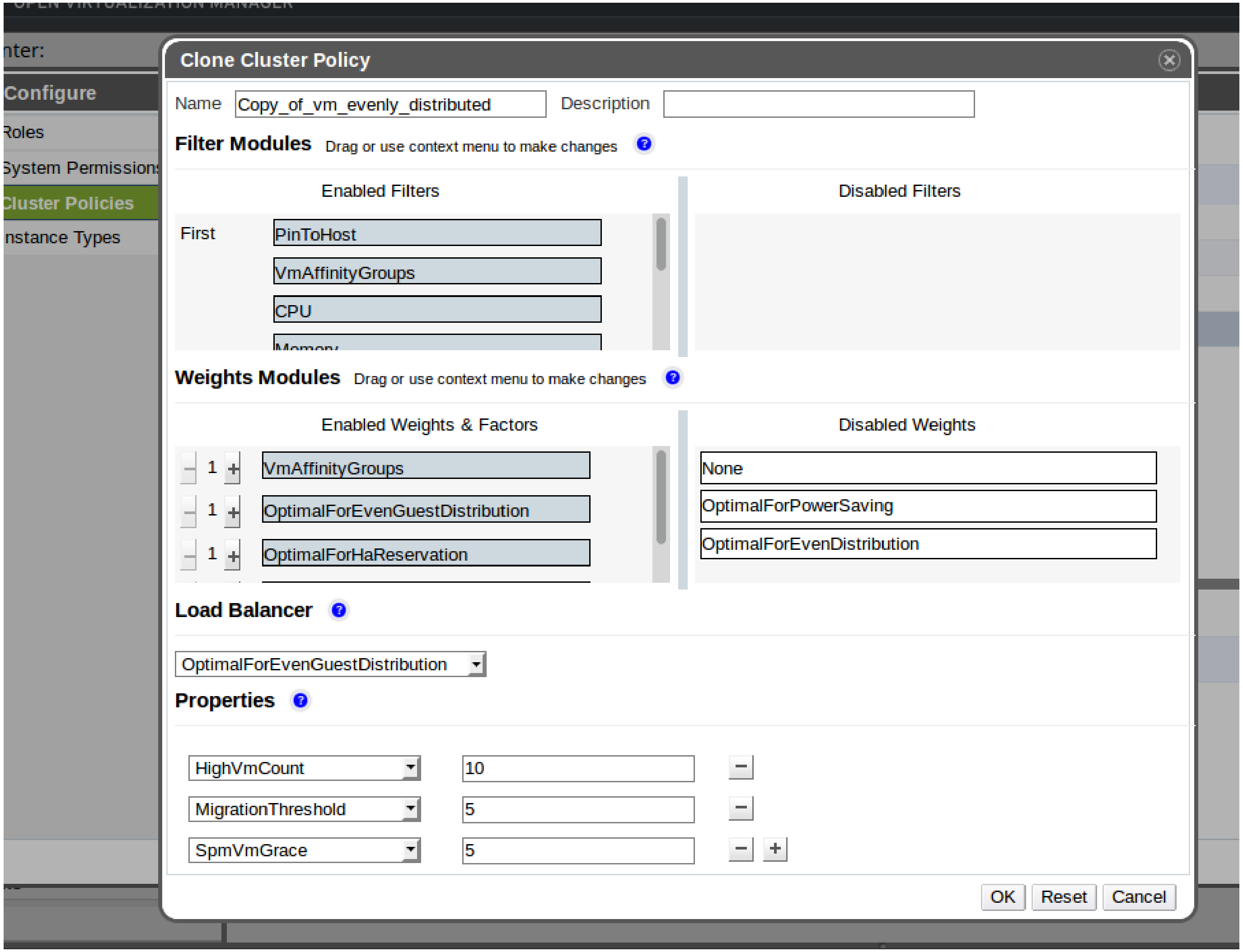
***Host 4 sum: 3*5+12*2 = 39**

Balancers

- Triggers a scheduled task to determine which VM needs to be migrated to one of under-utilized hosts
- A single load balancing logic is allowed per cluster

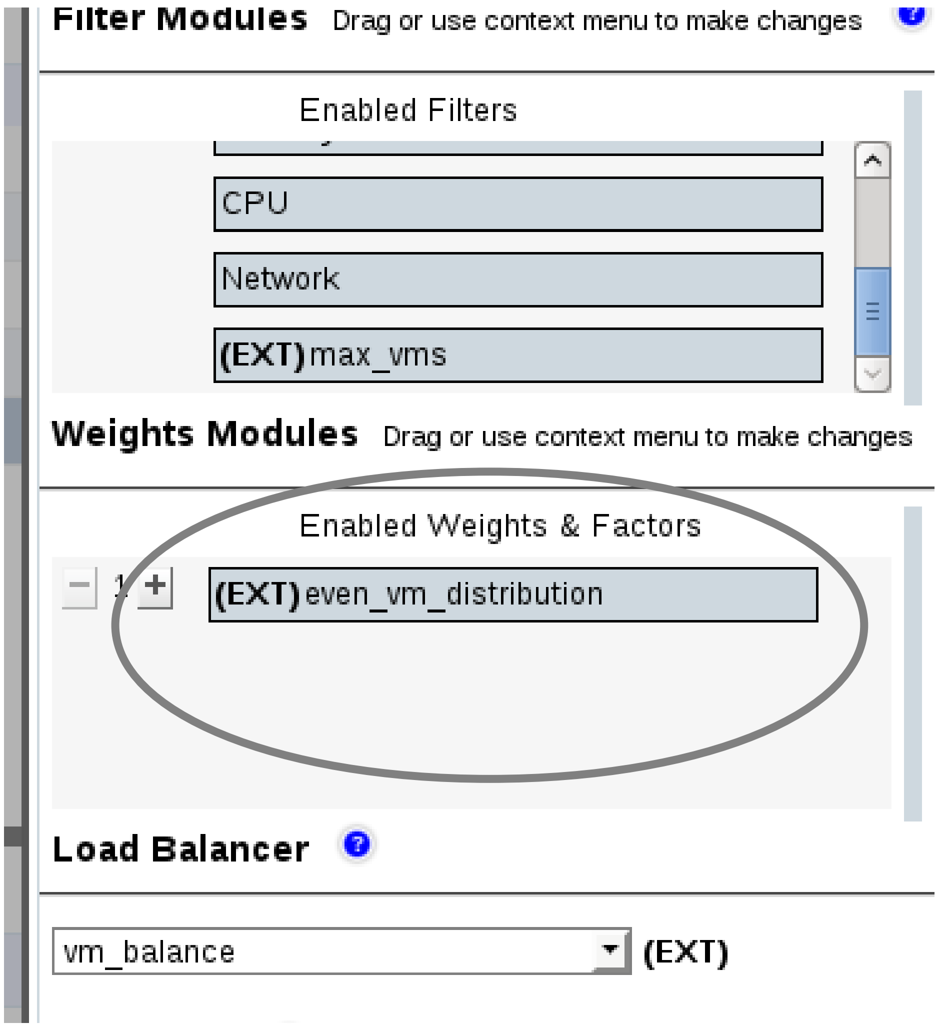


Filters, Weights, Balancers



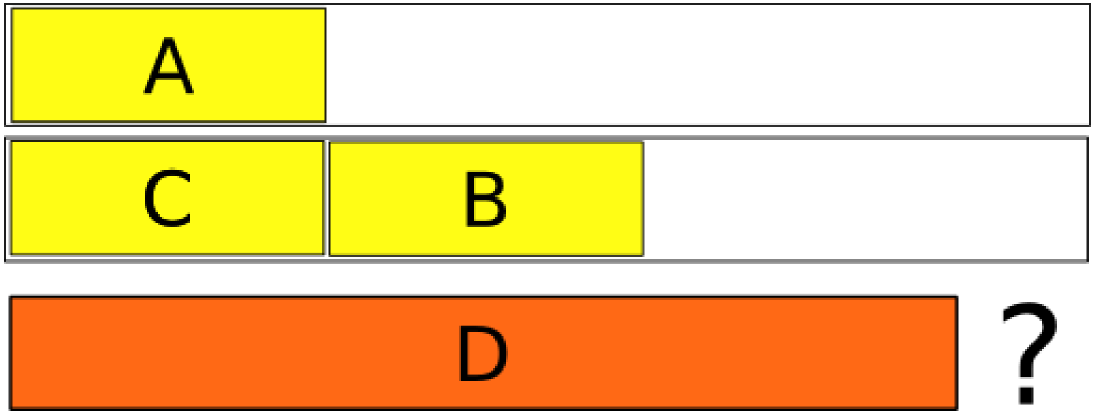
External Scheduler

- External service written in python and run as a separate process from the engine
- External service provides:
  - Engine safety
  - Should allow additional languages
  - Future option of scheduling as a service



Optimizer Goals

- Better load balancing
- Configurable by existing cluster policy
- Separate machine to protect ovirt-engine
- Starting a VM that can't be placed directly
  - Space needs to be created first



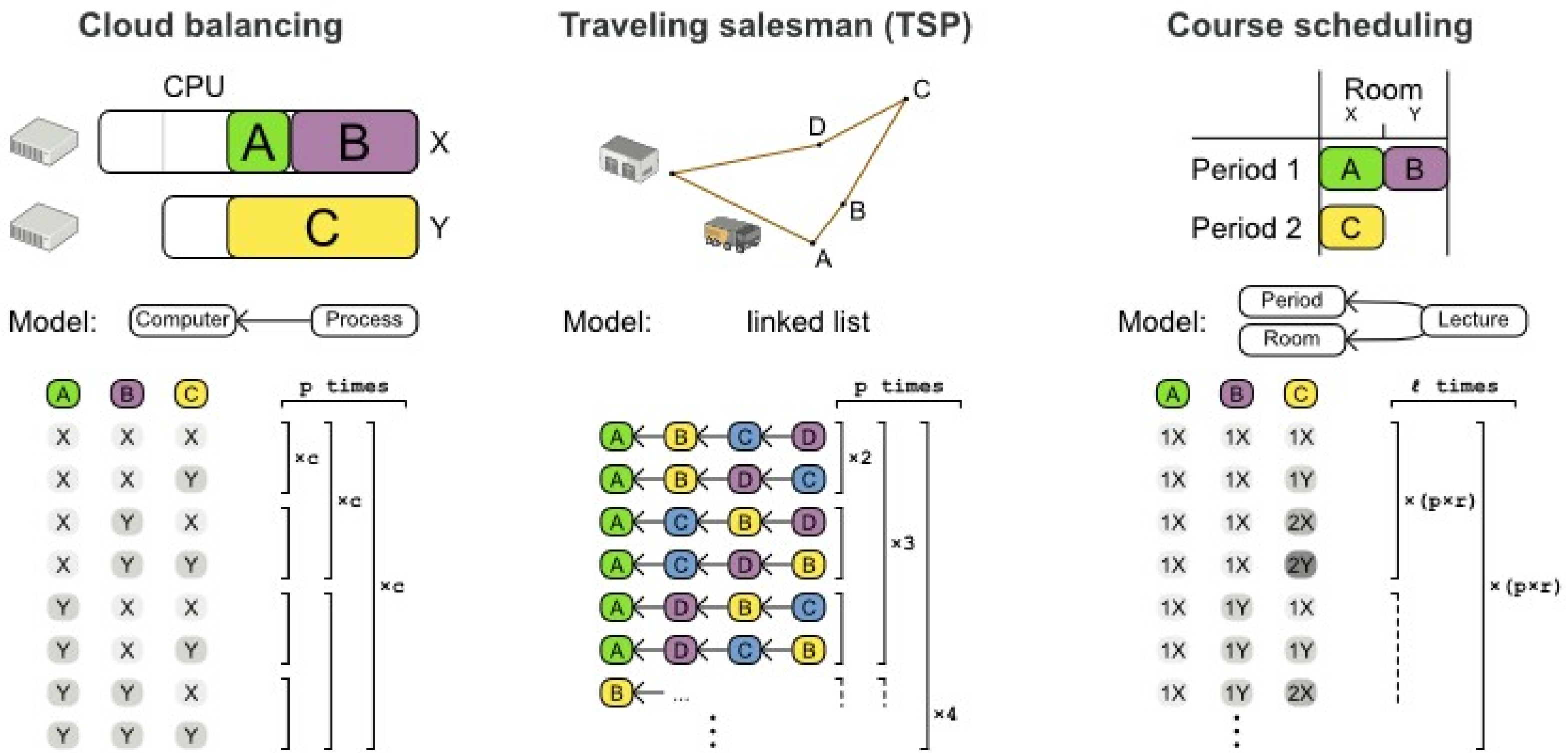
Machine reassignment problem

- Defined by set of machines and set of processes
- Each machine has some resources (CPU, RAM, ...)
- Each process requires resources
- NP-complete (variant of bin packing)
  - Easy to verify a given solution to a problem in reasonable time.
  - There is no silver bullet to find the optimal solution of a problem in reasonable time (*).

Big Problems

Calculate the size of the search space

Given a Solution model, how many different combinations can it represent?



Search space: c^p

# computers	# processes	search space
2	3	8
100	300	10^{600}
200	600	10^{1380}
400	1200	10^{6967}

Search space: $n!$

# customers	search space
4	24
100	10^{157}
1000	10^{2567}
10000	10^{35659}

Search space: $(p \times r)^{\ell}$

# periods	# rooms	# lectures	space
2	2	3	64
36	6	100	10^{233}
36	18	400	10^{1124}
36	36	800	10^{2490}

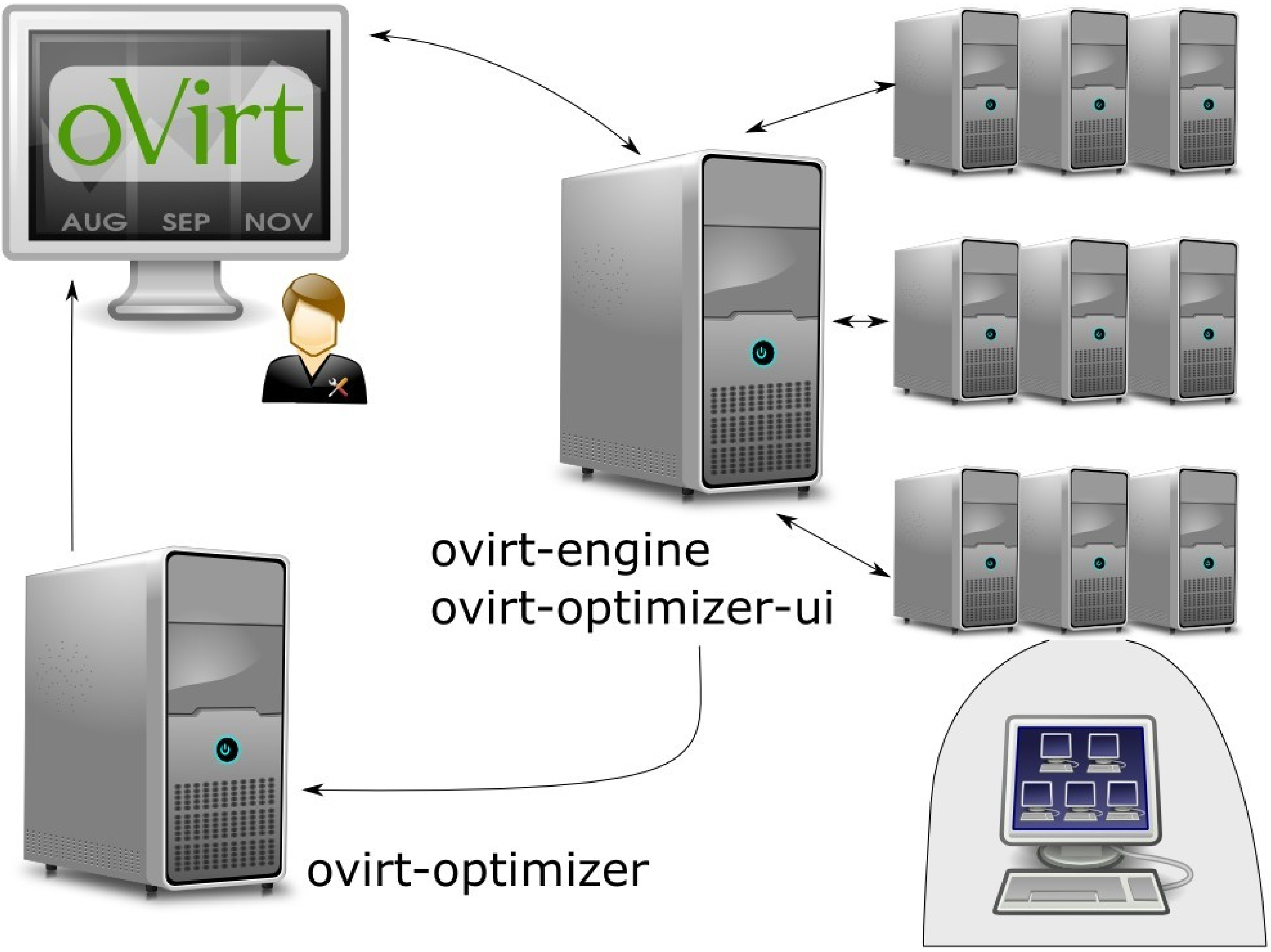
Enter OptaPlanner

- optaplanner.org
- Optimization engine
- Many search algorithms
- Uses Drools Rule Language (DRL) for scoring
  - drools.org

Probabilistic approach

- Random search
  - Randomly generate a candidate solution
  - Evaluate and assign a score
  - Accept if better than the current
  - Rinse and repeat
- Smarter than random
  - Simulated annealing – closer and closer neighbors
  - Tabu search – do not repeat mistakes

OptaPlanner and oVirt

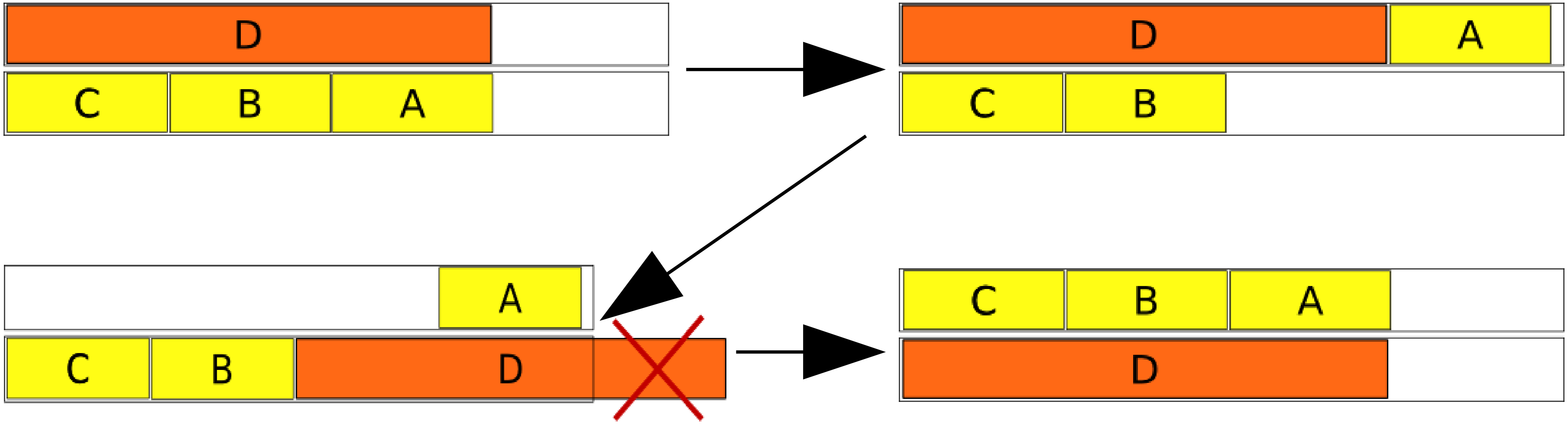


OptaPlanner and oVirt

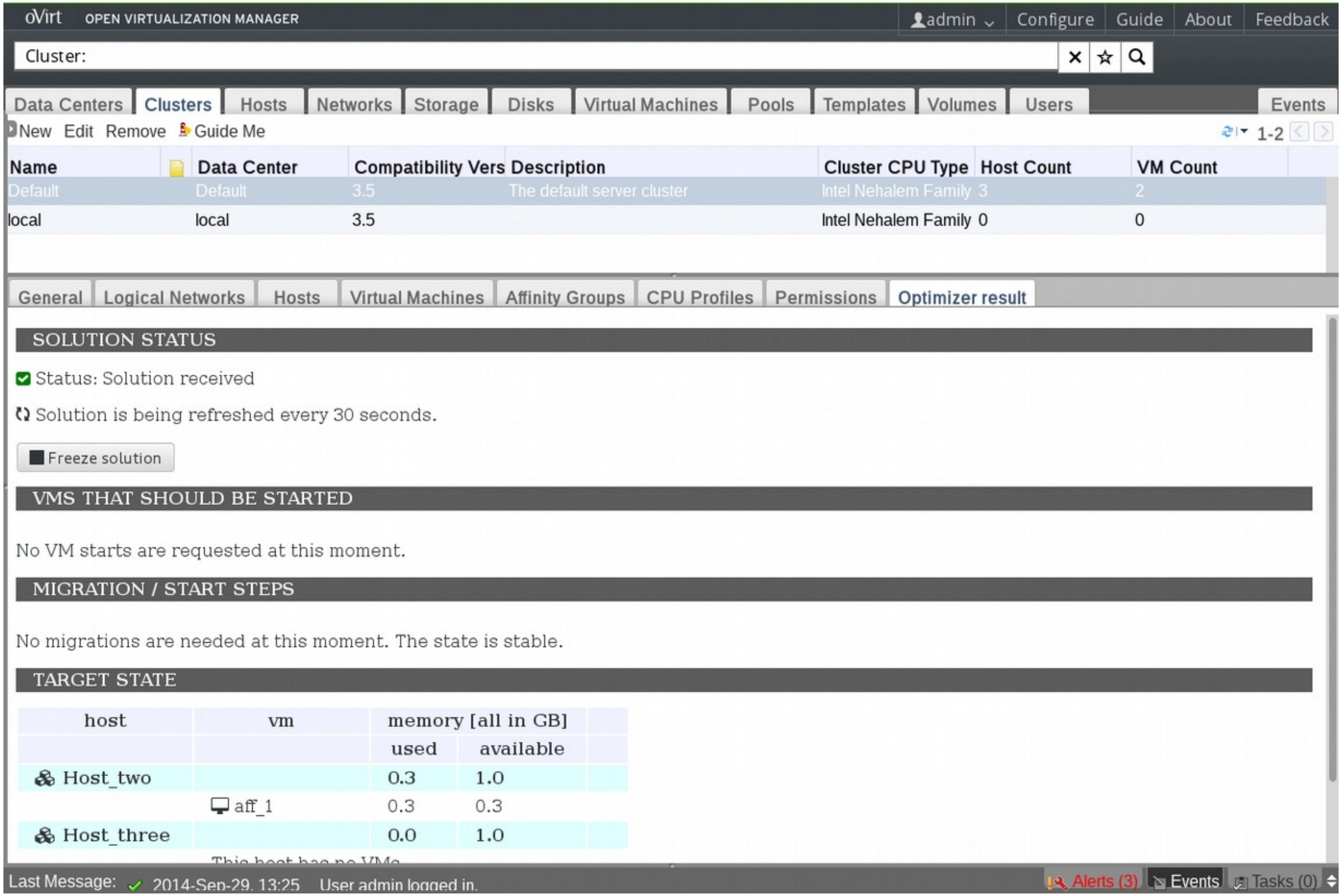
- oVirt's Java-based policy units converted to DRL-based rules in order to honor admin-set filters and weights
  - not all policy units yet available through API
    - hosted engine score filters
    - CPU load-based balancing
- cluster info periodically acquired by the optimizer over oVirt's REST API, converted, and fed to the OptaPlanner's fact database
- performance is improved by caching all rule matches
- All previous facts and rules are then used together by the OptaPlanner solver engine to compute the result.
- The optimizer service keeps running and improving the solution.
- When something in the cluster changes, the facts update and the solver resumes using the current best solution as a base point.

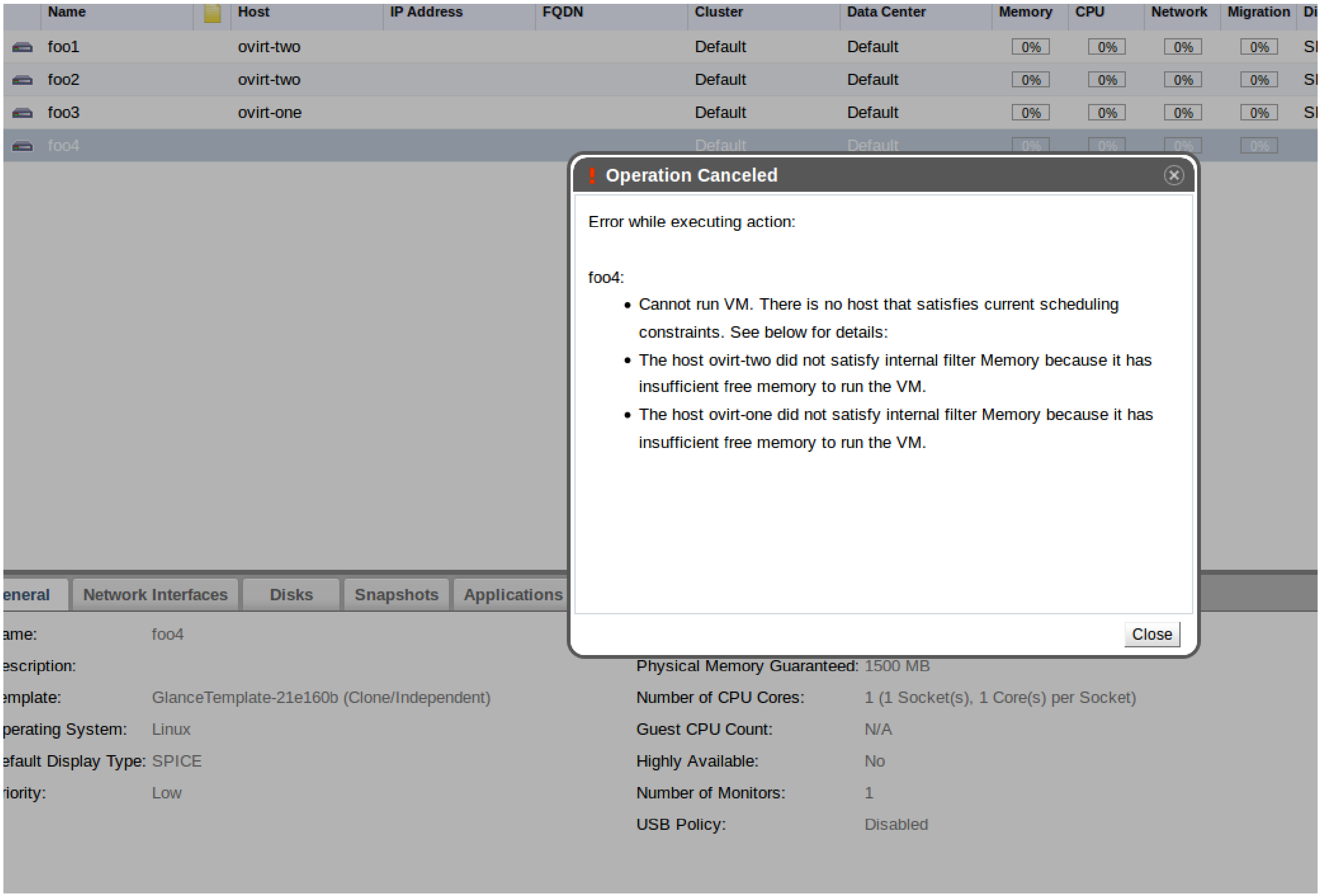
Optimization steps

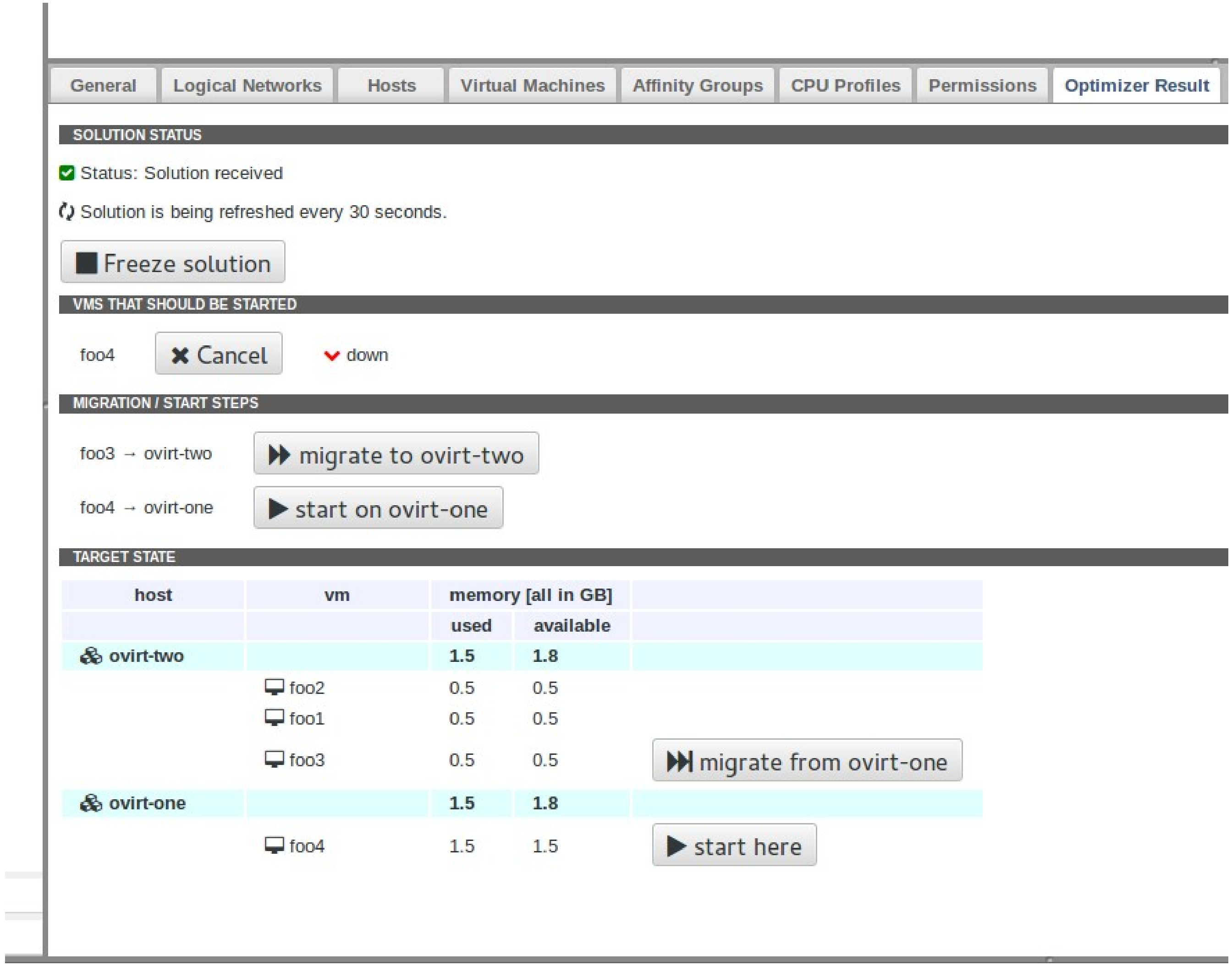
- Number of steps limited
- Slower to converge than simple “get me the optimum”
- Hard constraint check for each intermediate state
- Soft constraint check for the final situation only



Web admin integration




Name
Host
IP Address
FQDN
Cluster
Data Center
Memory
CPU
Network
Migration
foo1 ovirt-two Default Default 0% 0% 0% 0%
foo2 ovirt-two Default Default 0% 0% 0% 0%
foo3 ovirt-one Default Default 0% 0% 0% 0%
foo4 Default Default 0% 0% 0% 0%
Operation Canceled
Error while executing action:
foo4:
Cannot run VM. There is no host that satisfies current scheduling constraints. See below for details:
The host ovirt-two did not satisfy internal filter Memory because it has insufficient free memory to run the VM.
The host ovirt-one did not satisfy internal filter Memory because it has insufficient free memory to run the VM.
Close
Network Interfaces
Disks
Snapshots
Applications
foo4
GlanceTemplate-21e160b (Clone/Independent)
Linux
SPICE
Low
Physical Memory Guaranteed: 1500 MB
Number of CPU Cores: 1 (1 Socket(s), 1 Core(s) per Socket)
Guest CPU Count: N/A
Highly Available: No
Number of Monitors: 1
USB Policy: Disabled


General
Logical Networks
Hosts
Virtual Machines
Affinity Groups
CPU Profiles
Permissions
Optimizer Result
SOLUTION STATUS
Status: Solution received
Solution is being refreshed every 30 seconds.
Freeze solution
VMS THAT SHOULD BE STARTED
foo4
Cancel
down
MIGRATION / START STEPS
foo3 → ovirt-two
migrate to ovirt-two
foo4 → ovirt-one
start on ovirt-one
TARGET STATE
host
vm
memory [all in GB]
used
available
ovirt-two
1.5
1.8
foo2
0.5
0.5
foo1
0.5
0.5
foo3
0.5
0.5
migrate from ovirt-one
ovirt-one
1.5
1.8
foo4
1.5
1.5
start here

Looking Ahead

- Tighter integration with BRMS
- Full automation of the optimization
  - Using the optimizer instead of the internal scheduler in oVirt engine
- Support for more Policy Units
  - Custom DRL rules
  - Units coming from external scheduler
- Long term cooperation potential
  - OpenStack Gantt
  - Kubernetes
  - Mesos

Questions?

http://www.ovirt.org
jbrooks@redhat.com
@jasonbrooks
jbrooks on OFTC & Freenode IRC