ovirt-optimizer deep-dive

Probabilistic load balancing engine

29th of Oct 2014

Martin Sivák
Red Hat Czech

oVirt optimizer

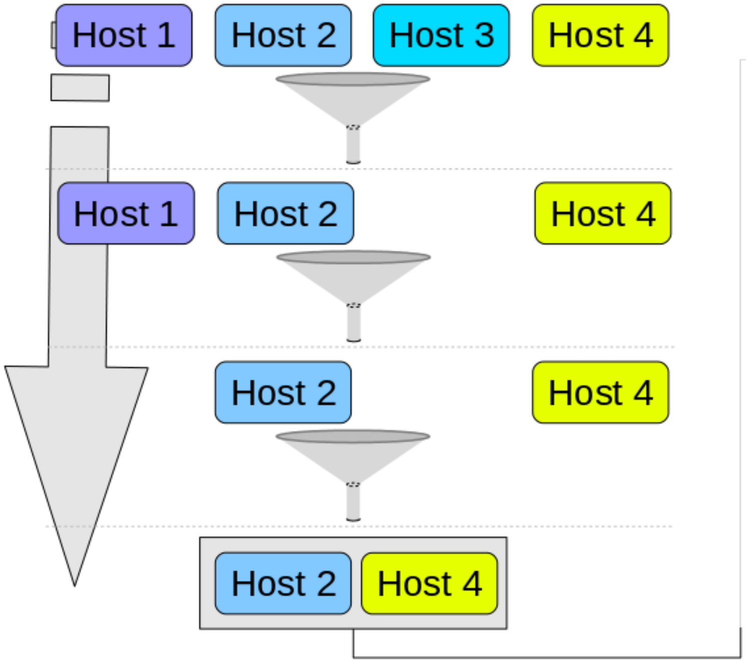
- Scheduling introduction
- Project goals
- Theory
- Optimization service details
- Demo

What is scheduling used for

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

oVirt way of computing host assignment

- Filters
- Weights
- Balancers



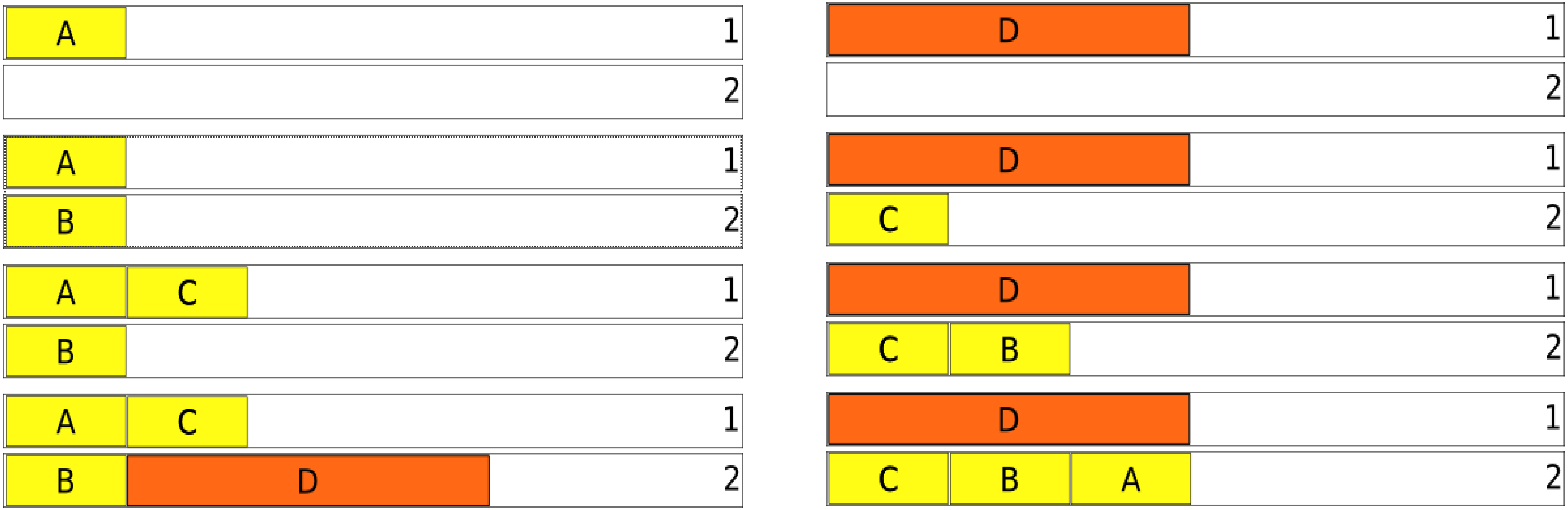
	func 1	func 2	sum
Factor	5	2	
Host 2	10	2	54
Host 4	3	12	**39***

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

- One-by-one
  - per cluster lock
  - wait_for_launch vs. starting
  - pending counters

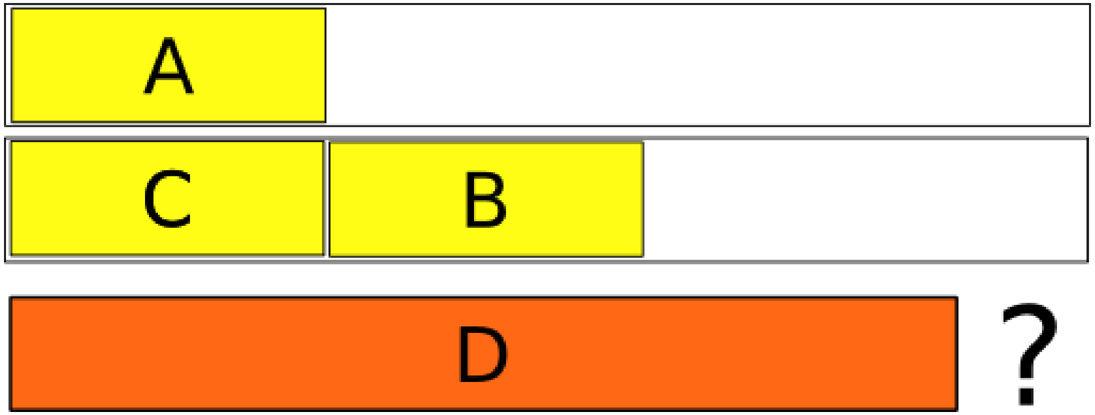
oVirt scheduling limitations

- One-by-one
- Load balancing
  - one per minute
  - select VM and candidate hosts



Goals

- Better load balancing
- Starting a VM that can't be placed directly
  - Space needs to be created first



- Configurable by existing cluster (migration) policy
- Separate machine to protect ovirt-engine

Machine reassignment problem

- Defined by set of machines and set of processes
- Each machine has some resources (CPU, RAM, ..)
- Each process requires resources
- **NP-hard** (variant of bin packing)

- Brute force is a no-go for any higher number of VMs
- We need reasonable response time

- http://challenge.roadef.org/2012/en/

Probabilistic approach

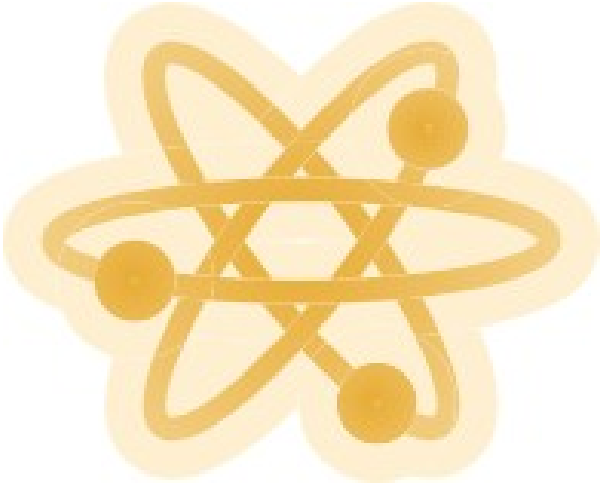
- Random search
  - Randomly generate a candidate solution
  - Evaluate and assign a score
  - Accept if better than the current
  - Rinse and repeat

- Simulated annealing – closer and closer neighbours
- **Tabu search** – do not repeat mistakes

- Genetic algorithms – natural selection
- ...

OptaPlanner and Drools

- http://www.drools.org/
- Fact database (KIE)
- Pattern matching rule evaluator
- Caching partial results

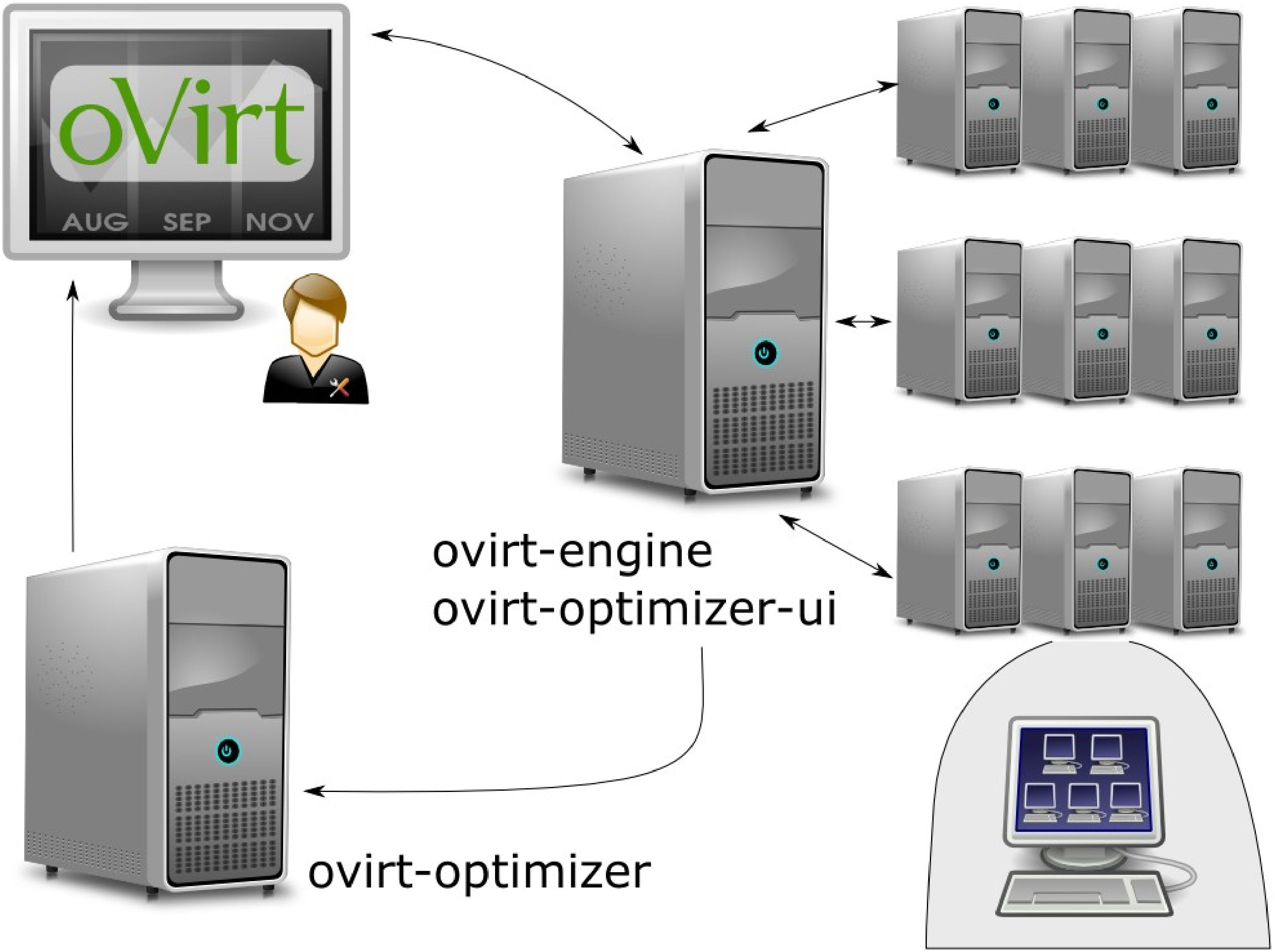


- http://www.optaplanner.org/
- Optimization engine
- Many search algorithms
- Uses DRL for scoring

Optimization as a service

- Constantly running service
  - One solver per cluster
- Real-Time planning
  - Pause condition – score has not improved in some time
- Receiving world facts updates
  - Query the ovirt-engine
  - Current status incorporated to the fact database
  - Solver restarted with the best solution as starting point

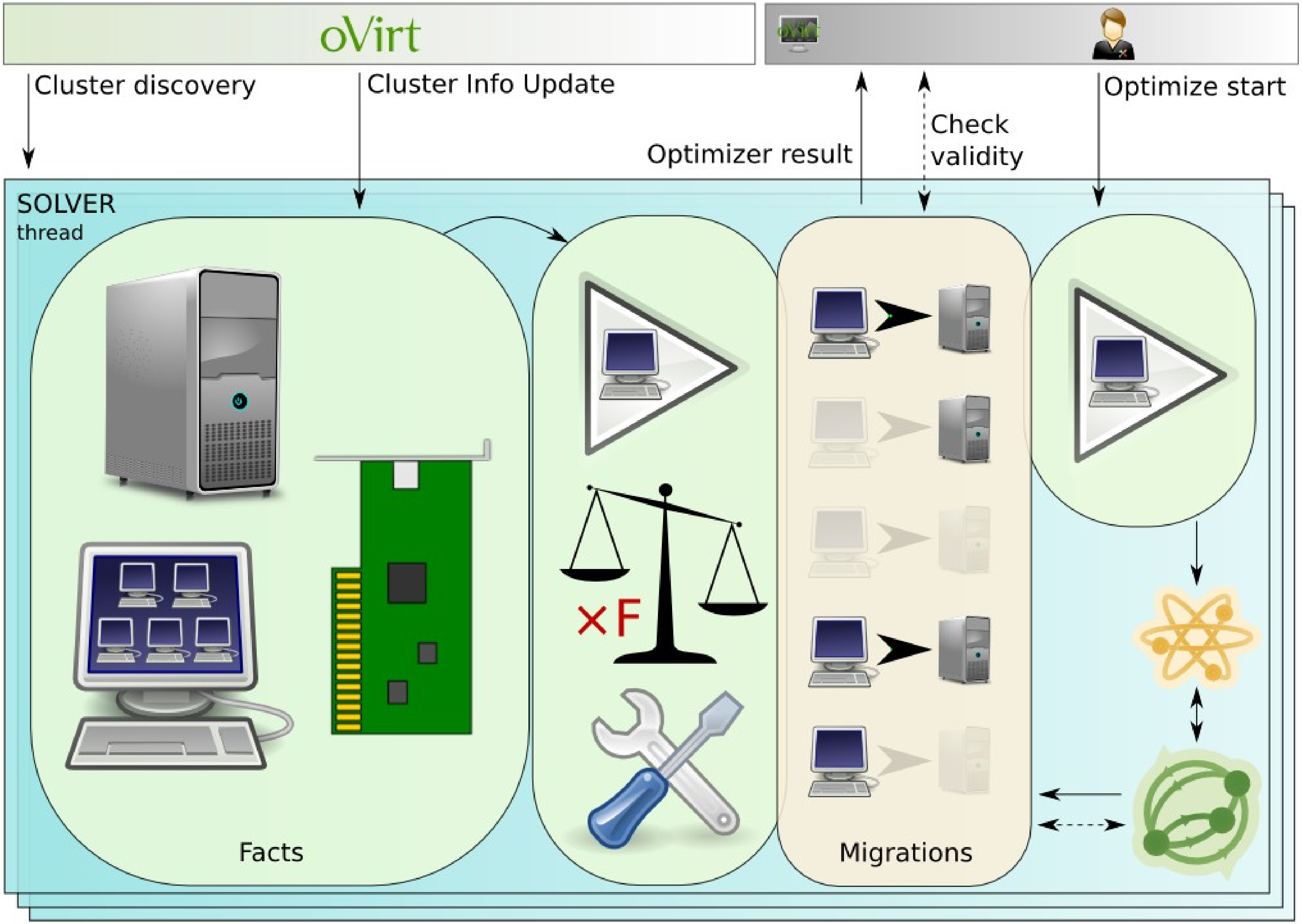
Architecture



ovirt-engine
ovirt-optimizer-ui

ovirt-optimizer

Internals


oVirt
Cluster discovery
Cluster Info Update
Optimizer result
Check validity
Optimize start
SOLVER
thread
×F
Facts
Migrations

Getting data from ovirt-engine

- Cluster discovery *(class ClusterDiscovery)*
  - cluster entity changes
  - start / stop solvers as needed
- Facts updates *(class ClusterInfoUpdater)*
  - list of Hosts, VMs, Networks, ...
  - enabled migration policy units
- Optimizations and issues
  - Subcollections – limit the amount of requests if possible
  - ID mapping – java object instance vs. cluster object

PolicyUnits vs. Drools rules

- PolicyUnits in ovirt-engine
  - Direct access to the engine DB
  - Complicated java algorithmic

- Drools rules in ovirt-optimizer
  - Pattern matching
  - Declarative and “easy” to read
  - Collections, sums of values, ...

Cluster Facts

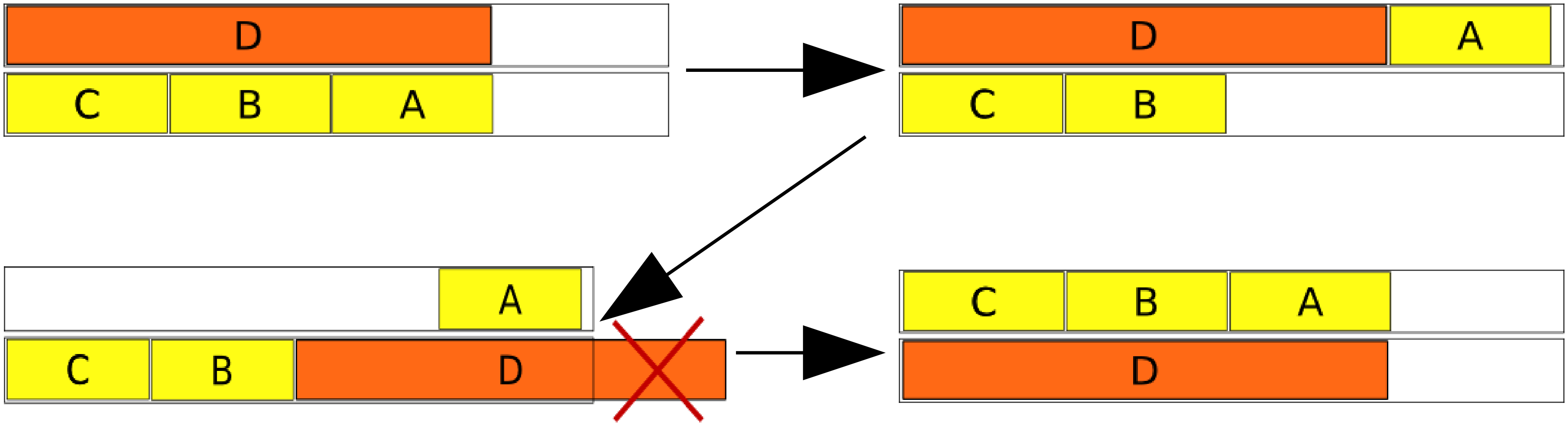
- REST entities → KIE fact database
- Supervised update cycle
  - OptaPlanner manages match cache and has to be notified of every updated or replaced entity
- Three fact sets
  - Cluster state facts, Configuration facts, User requests
- Some entities preprocessed
  - VmRunning, PolicyUnitEnabled
  - Improving rule readability (and cache performance)

Main planning entities

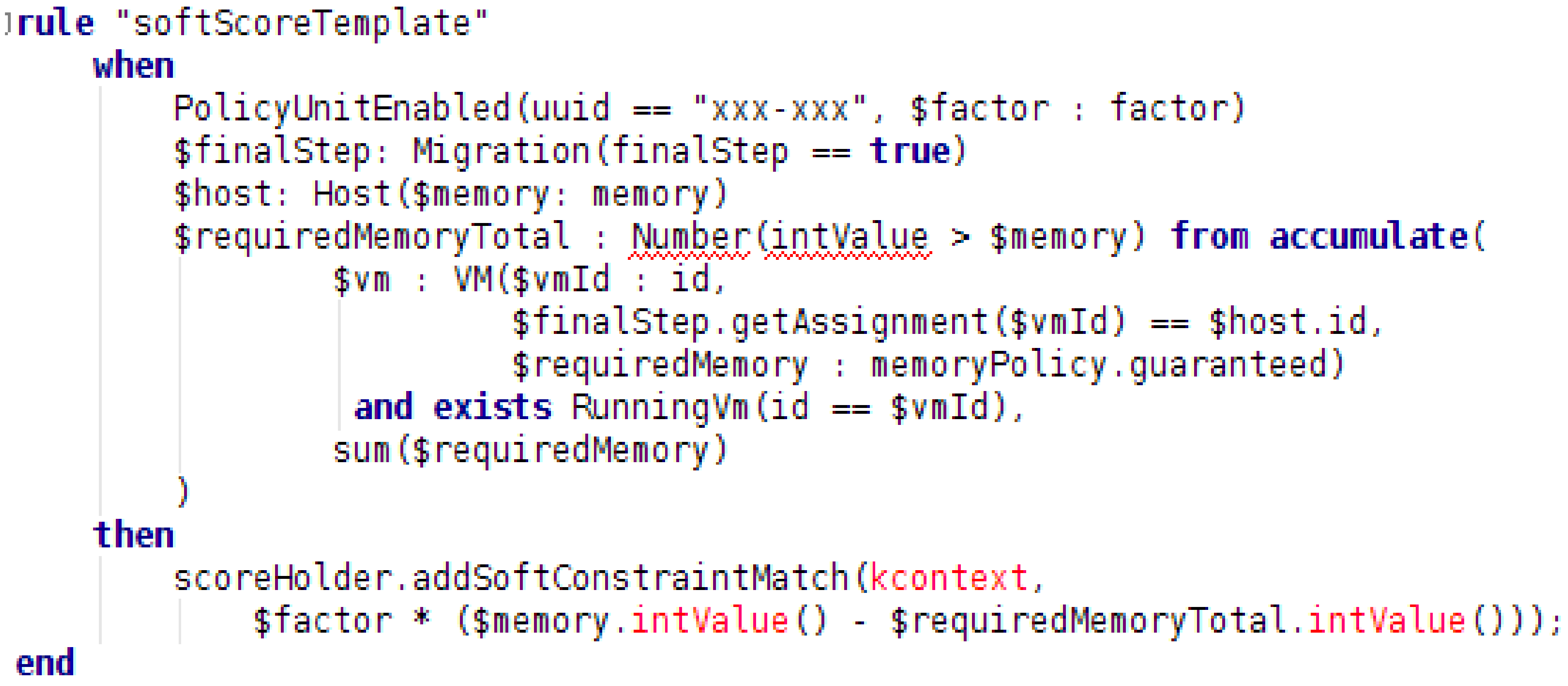
- OptimalDistributionStepsSolution
  - Represents a possible solution
  - Contains all facts about the cluster
- Migration
  - Represents one migration
  - Is linked to next and previous Migration entities
- MigrationStepChangeListener
  - Executed by Optaplanner when Migration changes
  - Recomputes cluster situation resulting from each Migration step to simplify hard constraint rules

Results – 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

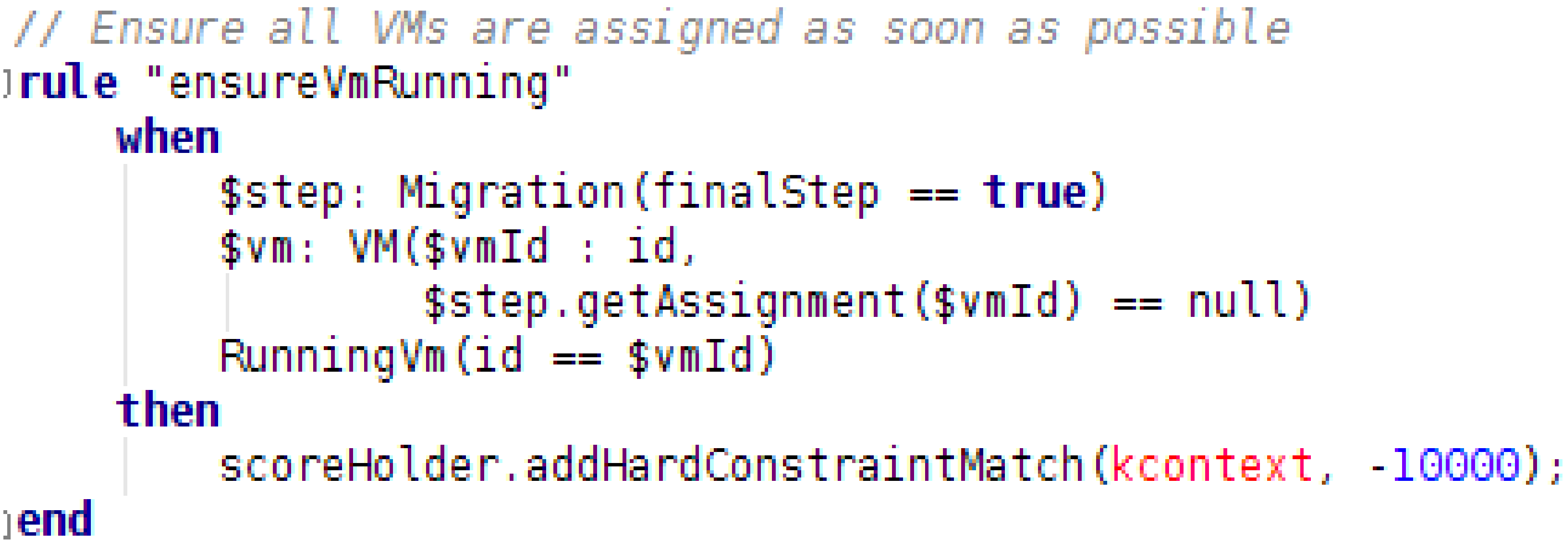


Soft constraint rule example



```
rule "softScoreTemplate"
    when
        PolicyUnitEnabled(uuid == "xxx-xxx", $factor : factor)
        $finalStep: Migration(finalStep == true)
        $host: Host($memory: memory)
        $requiredMemoryTotal : Number(intValue > $memory) from accumulate(
                $vm : VM($vmId : id,
                         $finalStep.getAssignment($vmId) == $host.id,
                         $requiredMemory : memoryPolicy.guaranteed)
                 and exists RunningVm(id == $vmId),
                sum($requiredMemory)
        )
    then
        scoreHolder.addSoftConstraintMatch(kcontext,
            $factor * ($memory.intValue() - $requiredMemoryTotal.intValue()));
end
```

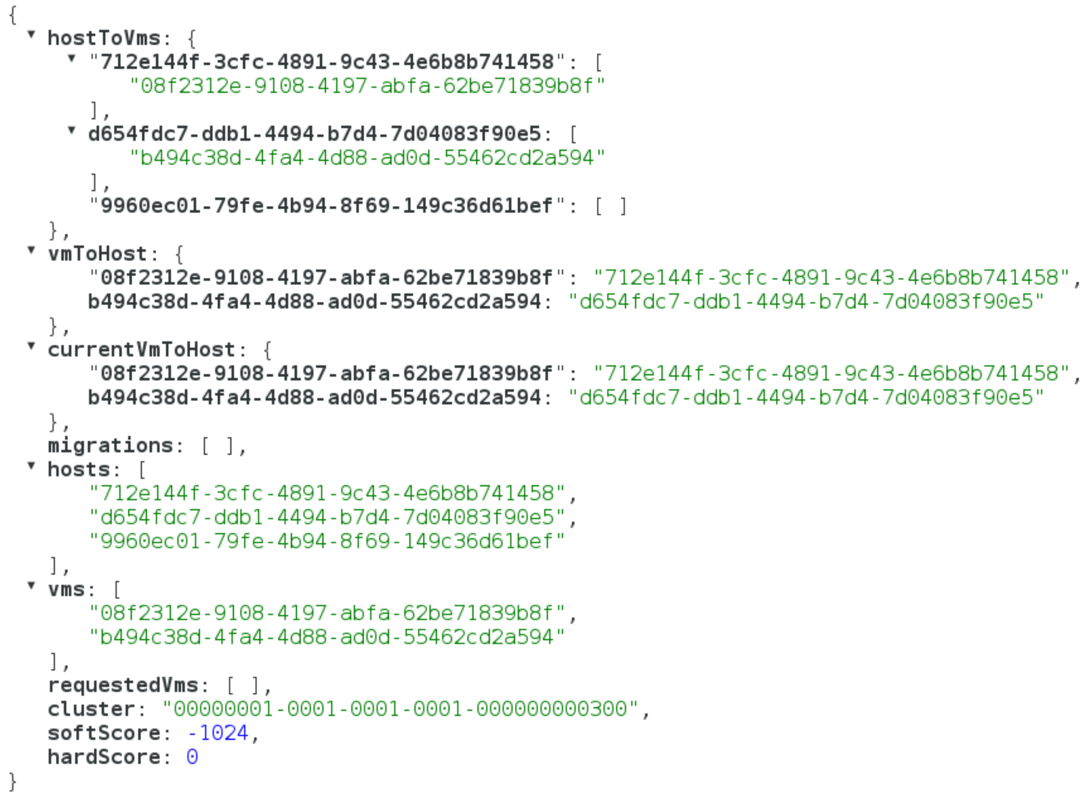
Hard constraint rule example



```
// Ensure all VMs are assigned as soon as possible
rule "ensureVmRunning"
    when
        $step: Migration(finalStep == true)
        $vm: VM($vmId : id,
                $step.getAssignment($vmId) == null)
        RunningVm(id == $vmId)
    then
        scoreHolder.addHardConstraintMatch(kcontext, -10000);
end
```

Reporting results

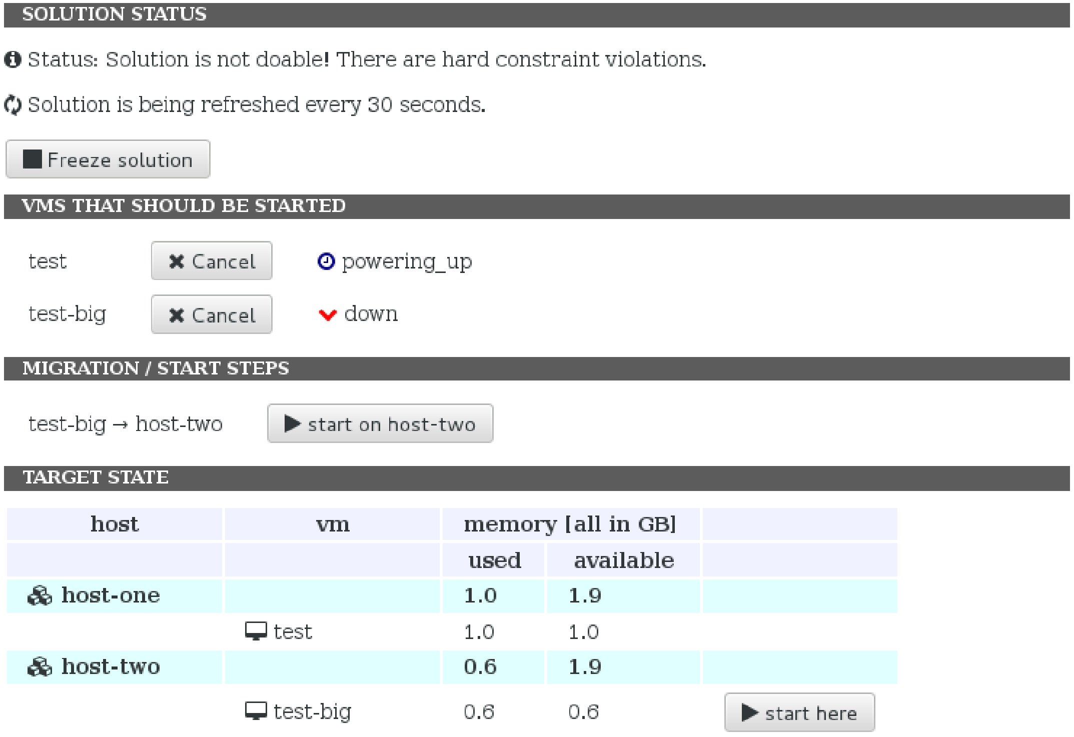
- One REST endpoint per cluster
  - GET /ovirt-optimizer/results/{clusterId}
- Result structure - json
  - IDs only
  - Current situation
  - Final situation
  - Steps
  - Start requests
  - Score



```
{
  ▾ hostToVms: {
      ▾ "712e144f-3cfc-4891-9c43-4e6b8b741458": [
            "08f2312e-9108-4197-abfa-62be71839b8f"
        ],
      ▾ d654fdc7-ddb1-4494-b7d4-7d04083f90e5: [
            "b494c38d-4fa4-4d88-ad0d-55462cd2a594"
        ],
        "9960ec01-79fe-4b94-8f69-149c36d61bef": [ ]
    },
  ▾ vmToHost: {
        "08f2312e-9108-4197-abfa-62be71839b8f": "712e144f-3cfc-4891-9c43-4e6b8b741458",
        b494c38d-4fa4-4d88-ad0d-55462cd2a594: "d654fdc7-ddb1-4494-b7d4-7d04083f90e5"
    },
  ▾ currentVmToHost: {
        "08f2312e-9108-4197-abfa-62be71839b8f": "712e144f-3cfc-4891-9c43-4e6b8b741458",
        b494c38d-4fa4-4d88-ad0d-55462cd2a594: "d654fdc7-ddb1-4494-b7d4-7d04083f90e5"
    },
    migrations: [ ],
  ▾ hosts: [
        "712e144f-3cfc-4891-9c43-4e6b8b741458",
        "d654fdc7-ddb1-4494-b7d4-7d04083f90e5",
        "9960ec01-79fe-4b94-8f69-149c36d61bef"
    ],
  ▾ vms: [
        "08f2312e-9108-4197-abfa-62be71839b8f",
        "b494c38d-4fa4-4d88-ad0d-55462cd2a594"
    ],
    requestedVms: [ ],
    cluster: "00000001-0001-0001-0001-000000000300",
    softScore: -1024,
    hardScore: 0
}
```

Webadmin integration – UI plugin

- Cluster optimization results
- VM names and info are obtained from engine's REST
  - Single request to get all VMs and second one for Hosts
  - Correlated with UUIDs from the solution



SOLUTION STATUS

Status: Solution is not doable! There are hard constraint violations.

Solution is being refreshed every 30 seconds.

■ Freeze solution

VMS THAT SHOULD BE STARTED

test	✖ Cancel	powering_up
test-big	✖ Cancel	down

MIGRATION / START STEPS

test-big → host-two ▶ start on host-two

TARGET STATE

host	vm	memory [all in GB]		
		used	available	
host-one		1.0	1.9	
	test	1.0	1.0	
host-two		0.6	1.9	
	test-big	0.6	0.6	▶ start here

Optimize start

- REST endpoints, both POST method

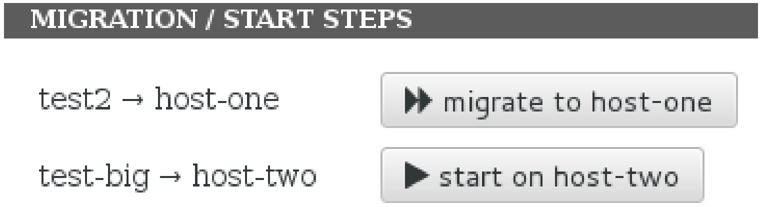
  /ovirt-optimizer/results/{clusterId}/request

  /ovirt-optimizer/results/{clusterId}/cancel

- VM's UUID passed in *cluster* request parameter

Applying the solution

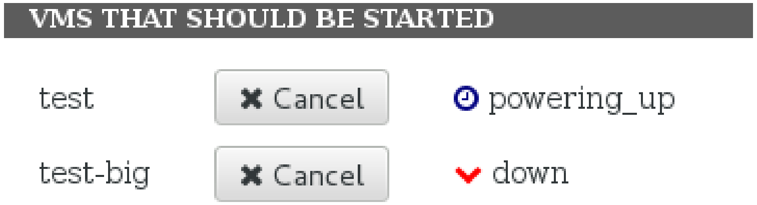
- Uses engine's REST in async mode to perform actions
- Only manual at this time
  - Hint for the administrator
  - Automatic in the future



MIGRATION / START STEPS

test2 → host-one	⏩ migrate to host-one
test-big → host-two	▶ start on host-two

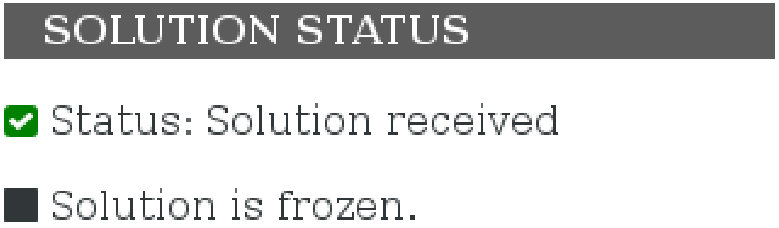
- Monitoring status



VMS THAT SHOULD BE STARTED

test	✖ Cancel	◷ powering_up
test-big	✖ Cancel	⌄ down

Solution freeze



SOLUTION STATUS

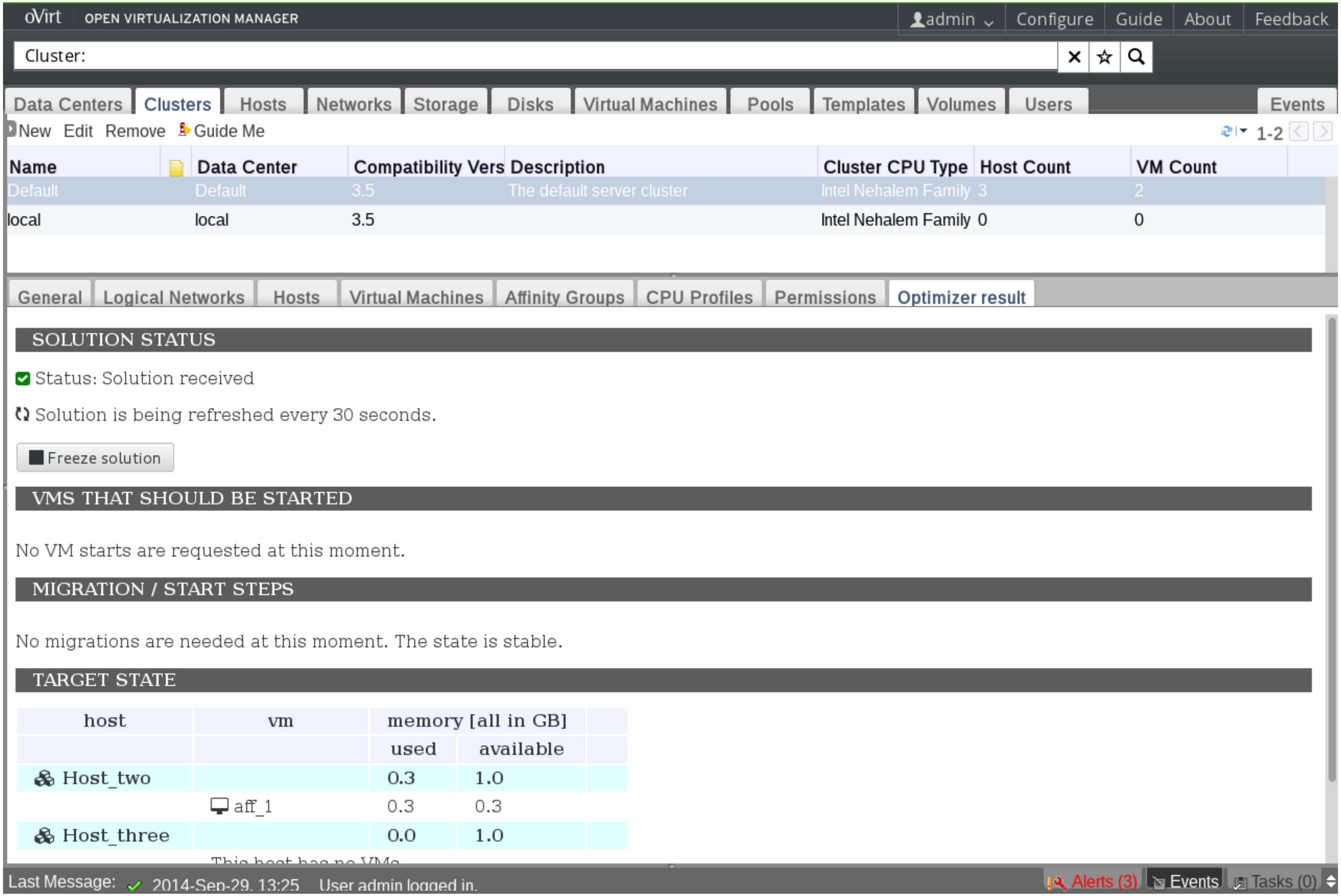
☑ Status: Solution received

■ Solution is frozen.

- Solution can change radically
- Manual actions are slow
- Freezing the solution refresh
- Validity monitoring
  - Another REST endpoint of the optimizer service

    /results/{clusterId}/score
  - Frozen solution submitted back to optimizer
  - Validity computed using the current facts
  - Hard and soft score returned back

Demo



Cluster:

Name		Data Center	Compatibility Vers	Description	Cluster CPU Type	Host Count	VM Count
Default		Default	3.5	The default server cluster	Intel Nehalem Family	3	2
local		local	3.5		Intel Nehalem Family	0	0

SOLUTION STATUS

☑ Status: Solution received

Solution is being refreshed every 30 seconds.

Freeze solution

VMS THAT SHOULD BE STARTED

No VM starts are requested at this moment.

MIGRATION / START STEPS

No migrations are needed at this moment. The state is stable.

TARGET STATE

host	vm	memory [all in GB]	
		used	available
Host_two		0.3	1.0
	aff_1	0.3	0.3
Host_three		0.0	1.0
	This host has no VMs		

Last Message: ✓ 2014-Sep-29. 13:25 User admin logged in.

Alerts (3) Events Tasks (0)

Future plans

- 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
- Review of OpenStack's Gantt, Kubernetics and Mesus
  - Possible cooperation, very long term

THANK YOU !

http://wiki.ovirt.org/wiki/Category:SLA
http://www.ovirt.org/Features/Optaplanner
users@ovirt.org
devel@ovirt.org

#ovirt irc.oftc.net