BTRPLACE

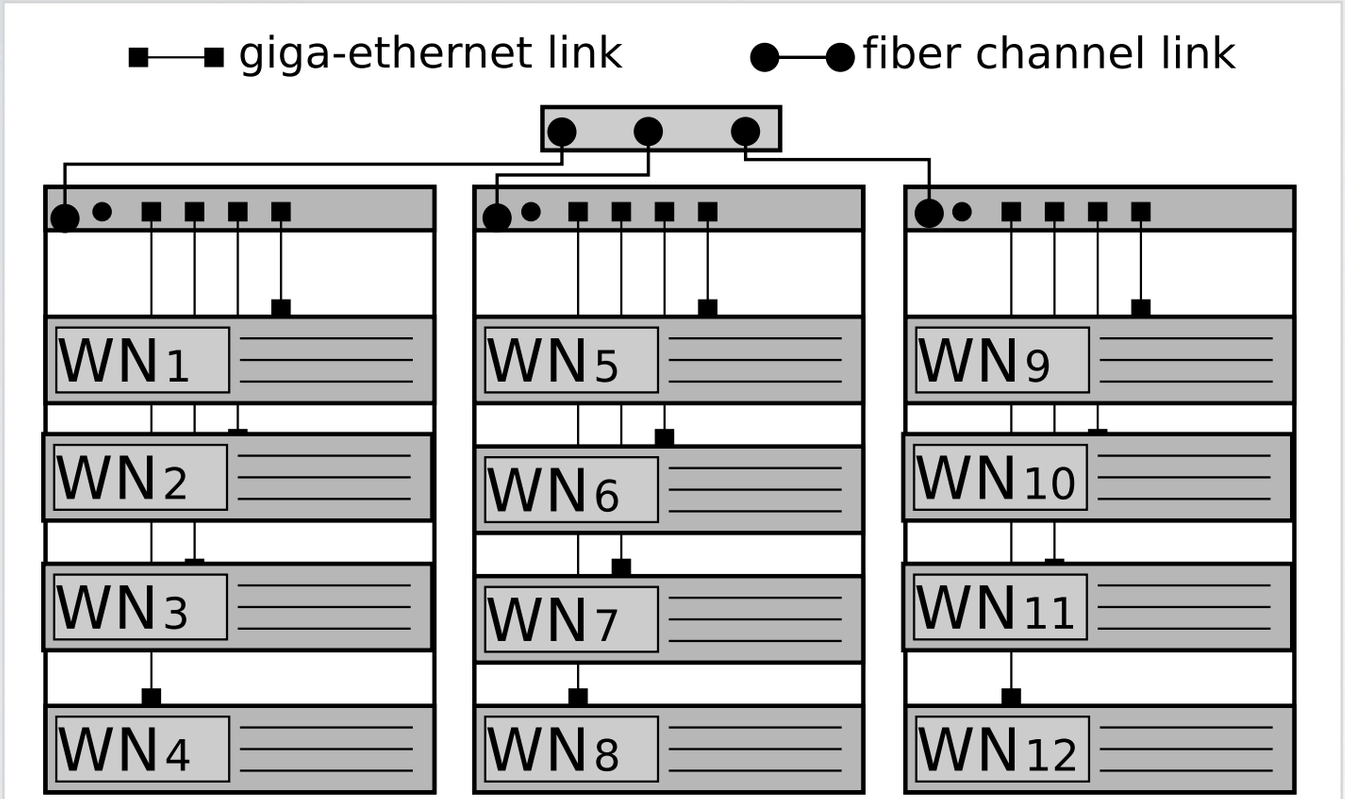
An extensible VM manager to face up to SLA expectations in a cloud

Research Team OASIS, INRIA/I3S

Fabien Hermenier
fabien.hermenier@unice.fr

SLA Management in Cloud @ Compas, 01/15/2013

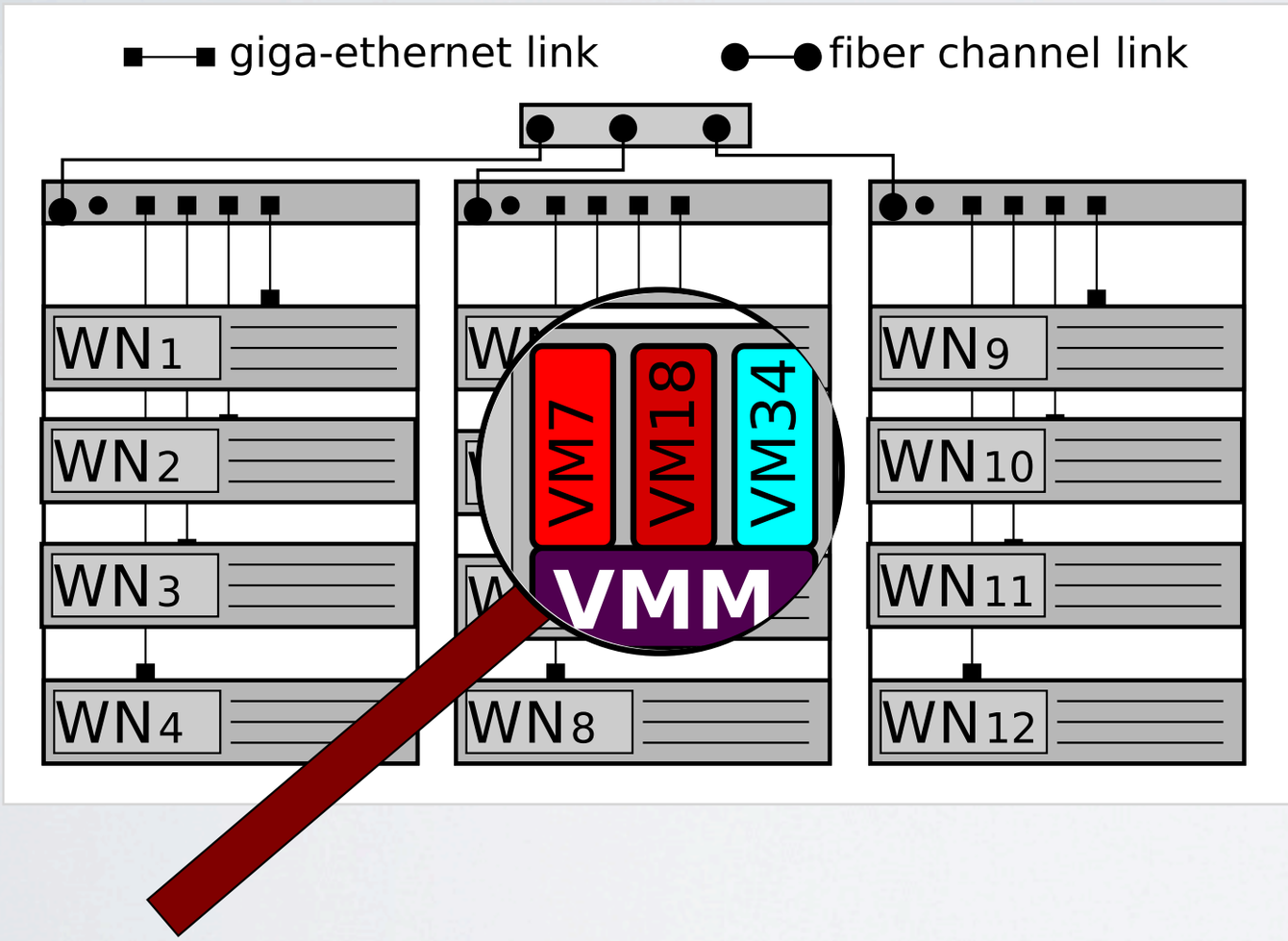
HOSTING PLATFORMS



Operators are looking for:

- manageability
- security
- efficient resource usage
- ...

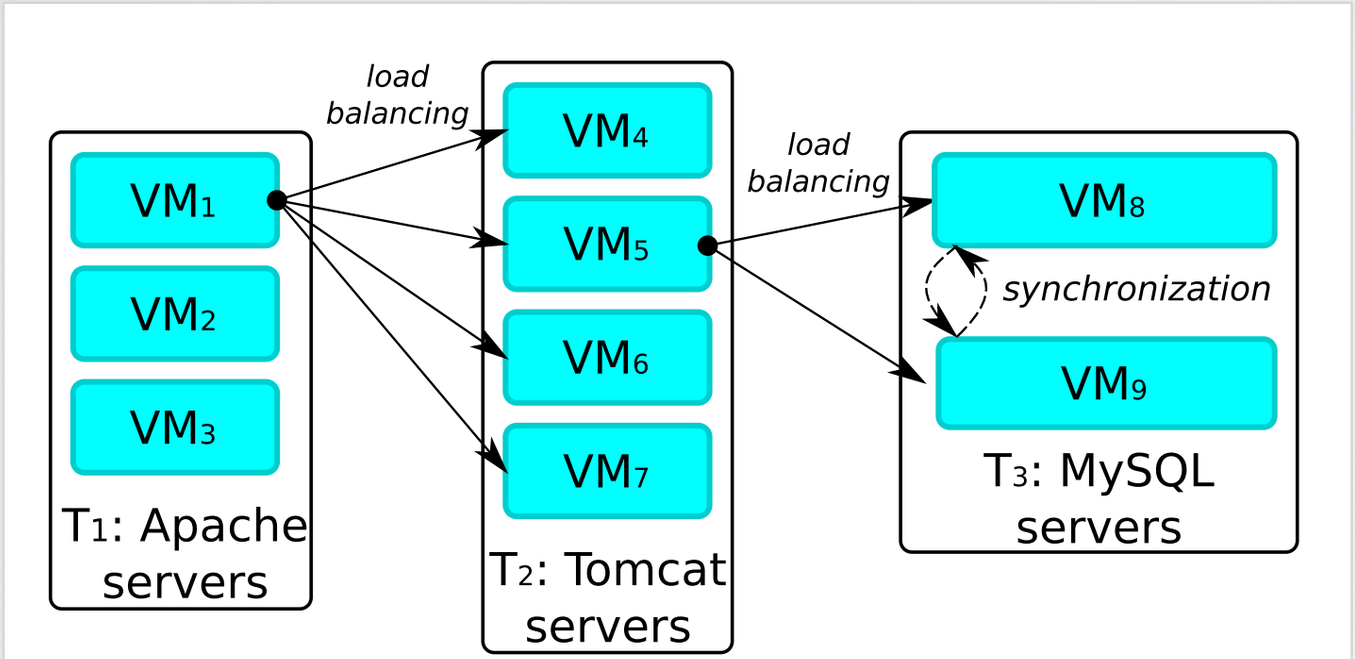
HOSTING PLATFORMS



Operators are looking for:

- manageability
- security
- efficient resource usage
- ...

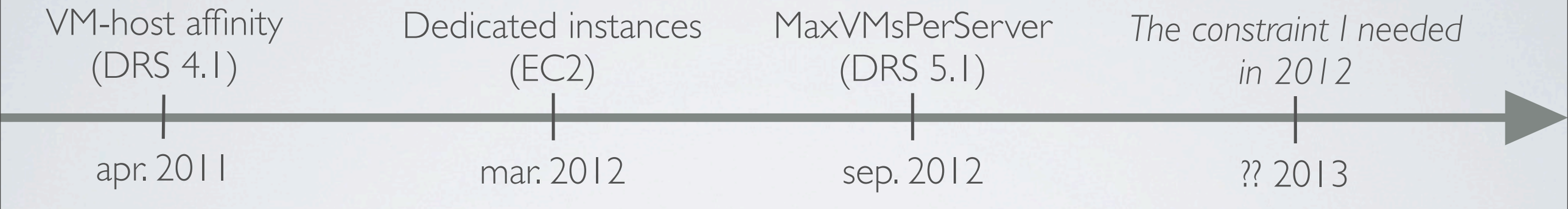
VIRTUAL APPLIANCE



Clients are looking for:

- performance
- reliability
- isolation
- ...

PLACEMENT CONSTRAINTS



- SLAs at the infrastructure level
- a unachieved story in which users are not the heroes
- current algorithms are not extensible by design

A CUSTOMIZABLE PLACEMENT ALGORITHM ?

Some problems :

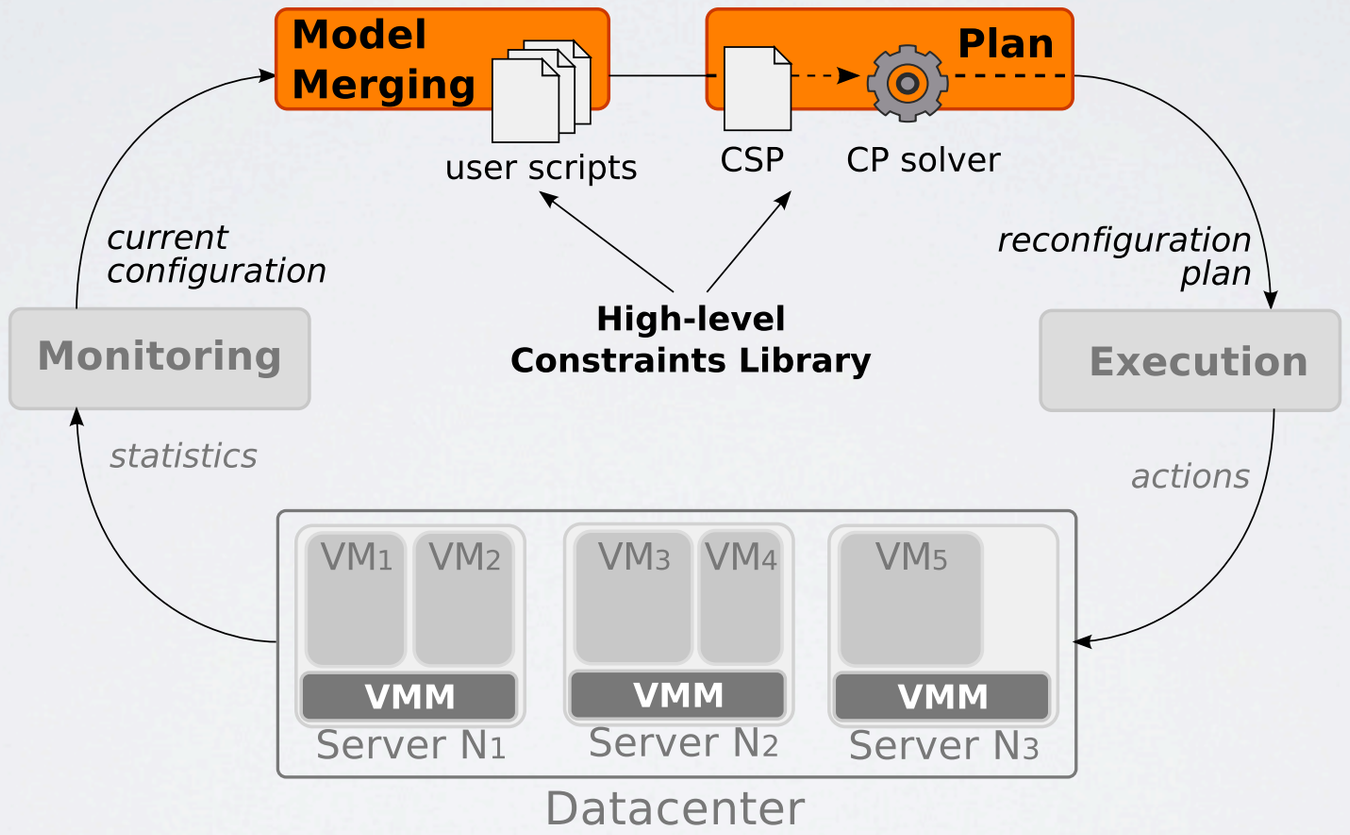
- constraints expressed by non-expert users
- numerous specific placement constraints
- concurrent placement constraints

One proposition :

- an extensible library of high-level placement constraints
- a composable VM placement algorithm

BTRPLACE

A customizable VM placement algorithm



✓ configurable

✓ composable

CONFIGURATION SCRIPTS

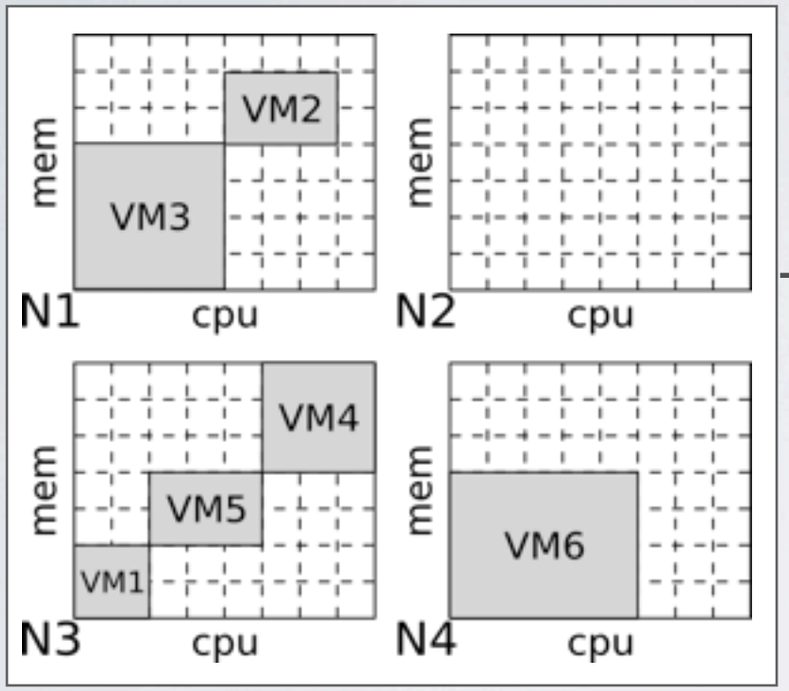
```
namespace datacenter;

$servers = @N[1..12];
$racks = {@N[1..4],@N[5..8],@N[9..12]};

export $racks to *;
```

```
namespace sysadmin;
import datacenter;
import client.*;

vmBtrplace: large;

fence(vmBtrplace, @N1);
lonely(vmBtrplace);
ban($clients, @N5);
```

```
namespace clients.app1;
import datacenter;

VM[1..7]: small<clone, boot=5,halt=5>;
VM[8..10]: large<clone, boot=60,halt=10>;
$T1 = {VM1, VM2, VM3};
$T2 = VM[4..7];
$T3 = VM[8,10];

for $t in $T[1..3] {
    spread($t);
}

among($T3,$racks);
export $me to sysadmin;
```

- provide datacenter and appliances descriptions
- human friendly definition of a viable datacenter

SAMPLE RECONFIGURATION

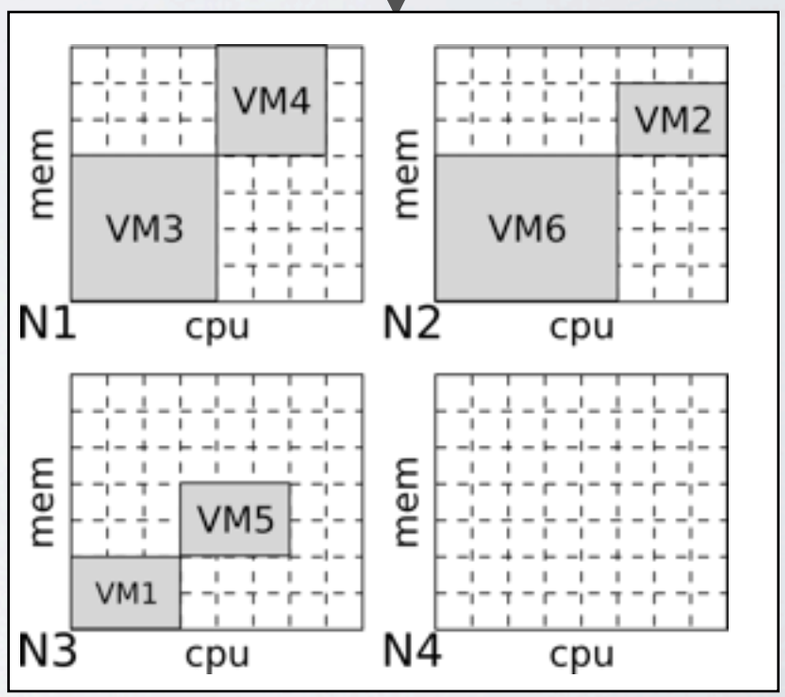



```
spread({VM3,VM2});
preserve({VM1},'ucpu', 3);
offline(@N4);
```

Btrplace

The reconfiguration plan :

```
0'00 to 0'02: relocate(VM2,N2)
0'00 to 0'04: relocate(VM6,N2)
0'02 to 0'05: relocate(VM4,N1)
0'04 to 0'08: shutdown(N4)
0'05 to 0'06: allocate(VM1,'cpu',3)
```

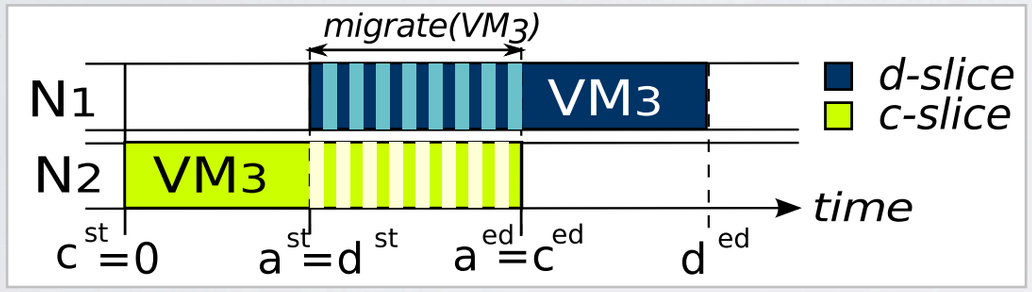


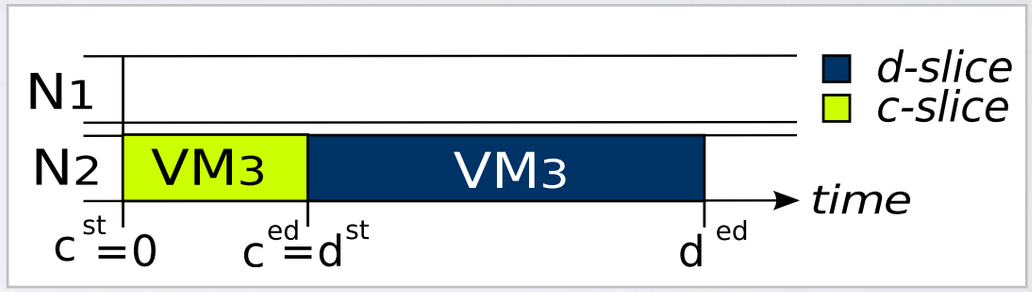
IMPLEMENTATION

- the core-RP models the VMs placement wrt. their resource usage
- placement constraints are interpreted to specialize the core-RP
- an implementation based on constraint programming
  - deterministic composition
  - high expressivity
  - the model is the implementation

MODELING CORE-RP

- actions are modeled wrt. their impact on resources using slices





- to place the d-slices: 2 *bin-packing* constraints
- to schedule the slices: a home-made *cumulatives*

MODELING THE PLACEMENT CONSTRAINTS

Using variables of the core-RP :

Variables related to VM Management	
c^{host}	Current host of the VM (constant)
c^{mem}, c^{cpu}	Current amount of memory and uCPU resources allocated to the VM (constant)
c^{ed}	Time the VM may leave its current host
d^{host}	Next host of the VM
d^{mem}, d^{cpu}	Next amount of memory and uCPU resources to allocate to the VM
d^{st}	Time the VM arrives on its next host
Variables related to server management	
n^q	Next state of the server
Variables related to action management	
a^{st}, a^{ed}	Times an action starts and ends, respectively

`spread({VM1,VM2})`:

$$allDifferent(d_1^{host}, d_2^{host}) \wedge$$
$$d_1^{host} = c_2^{host} \rightarrow d_1^{st} \geq c_2^{ed} \wedge$$
$$d_2^{host} = c_1^{host} \rightarrow d_2^{st} \geq c_1^{ed}$$

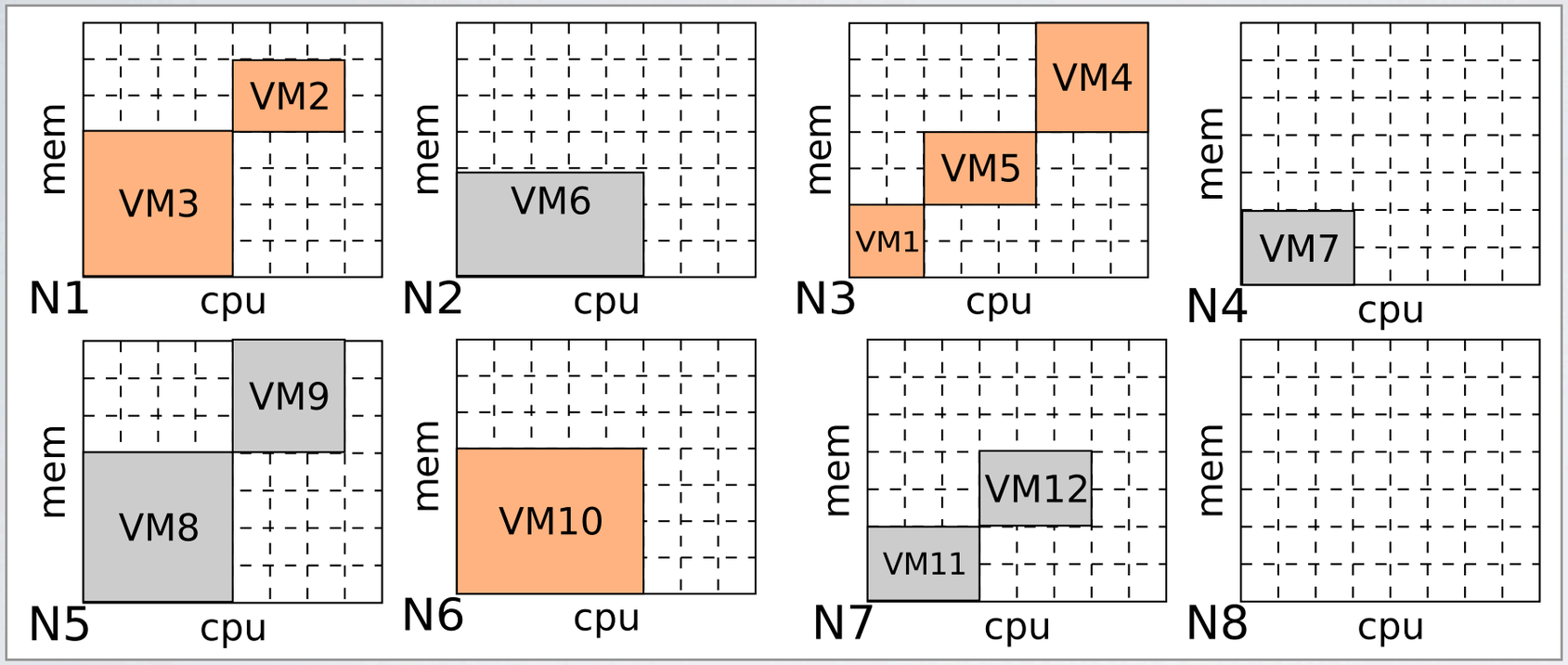
THE CONSTRAINTS LIBRARY

Initially : `spread, gather, among, splitAmong, ban, fence, lonely, quarantine, capacity, preserve, root, offline, oversubscription, noIdles`

Pending : `overbook, sequentialVMTransitions, maxOnlineNodes singleRunningCapacity, singleResourceCapacity, onlines, cumulatedResourceCapacity, maxSpareResources, minSpareResources, ...`

- multiple concerns: performance, isolation, reliability, administration, ...
- manipulate servers state, VM placement, resource allocation, action schedule

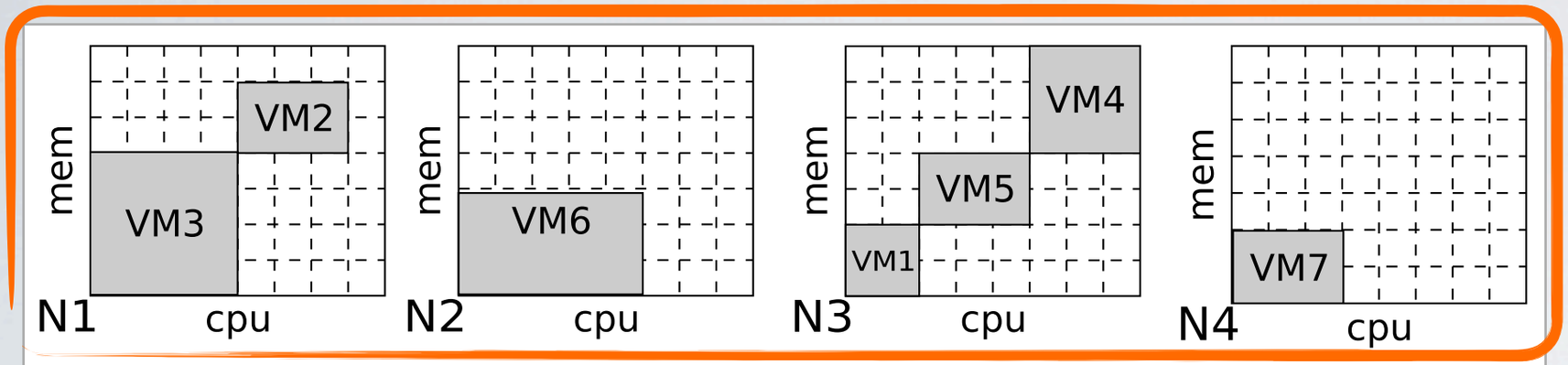
OPTIMIZING THROUGH *FILTER*

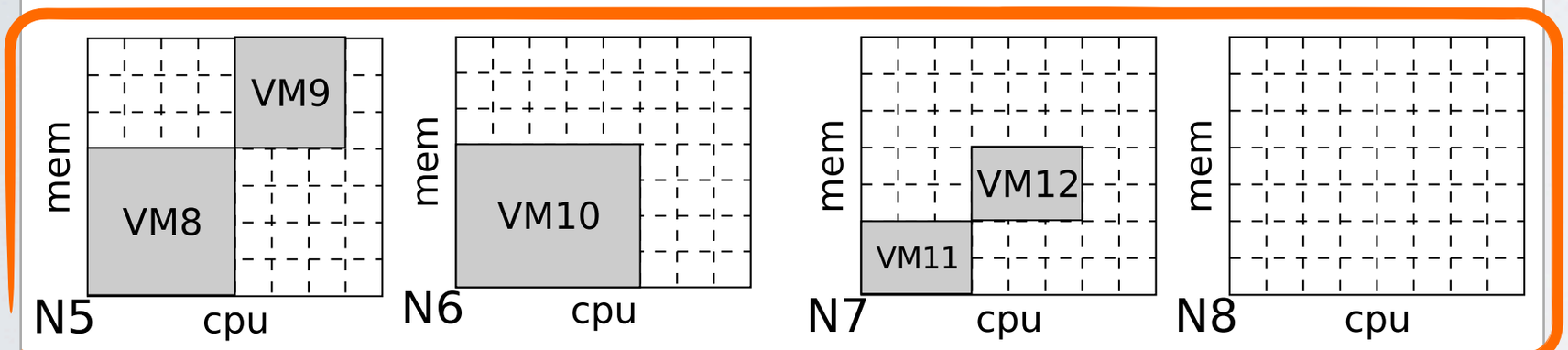



```
spread({VM3,VM2,VM8});
lonely({VM7});
preserve({VM1},'ucpu', 3);
offline(@N6);
ban($ALL_VMS,@N8);
fence(VM[1..7],@N[1..4]);
fence(VM[8..12],@N[5..8]);
```

- focus only on supposed mis-placed VMs
- provide RPs with less VMs to manage
- beware of under estimations !

OPTIMIZING TROUGH PARTITIONING






```
spread({VM3,VM2,VM8});
lonely({VM7});
preserve({VM1},'ucpu', 3);
offline(@N6);
ban($ALL_VMS,@N8);
fence(VM[1..7],@N[1..4]);
fence(VM[8..12],@N[5..8]);
```

- constraints may introduce independent RPs
- provide smaller RPs, solvable in parallel
- beware of resource fragmentation !

EVALUATION

- is Brplace flexible in practice ?
- does Btrplace makes the VMs placement reliable ?
- a *complete* approach for large problems, really ?

EXPRESSIVITY

The current library :

- covers VMWare DRS and EC2 placement constraints
- provides new relevant placement constraints

EXTENSIBILITY

Constraints implementation :

- concise: +/- 30 loc. per constraint
- «fast» to implement for an experienced user
- Fit4Green EU projects : un-experienced users of Btrplace

BTRPLACE EASES SERVER MAINTENANCE

8 servers run HA 3-tiers appliances

Time	Event	Reconfiguration Plan
2'10	+ban({WN8})	3 + **3 relocations** in 0'42
4'30	+ban({WN4})	2 + **7 relocations** in 1'02
7'05	-ban({WN4})	no reconfiguration
11'23	+ban({WN4})	**no solution**
11'43	-ban({WN8}) +ban({WN4})	2 relocations in 0'28

Btrplace prevented the mis-reconfigurations

SCALABILITY

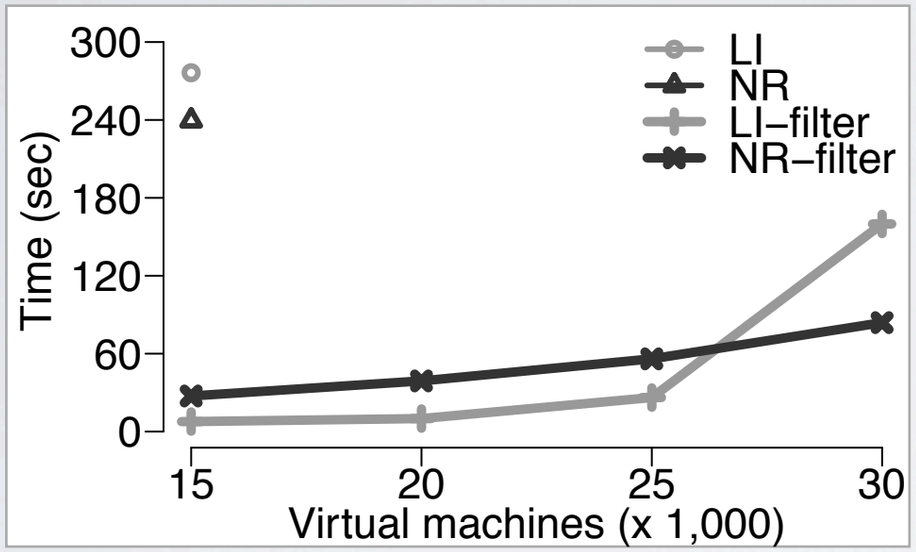
A simulated datacenter :

- 5,000 servers
- up to 1,700 3-tiers appliances (30,000 VMs)
- a resource usage up to 73%

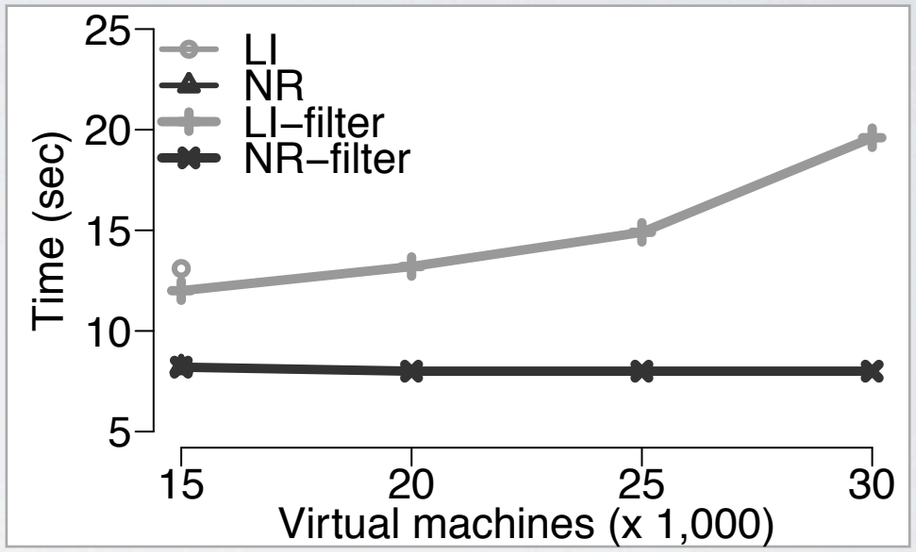
2 scenario:

- Load Increase (LI): 10% of the applications ask for 30% more uCPU
- Network Rewiring (NR): 5% of the servers are turned off for a network maintenance

THE *FILTER* OPTION



Solving duration

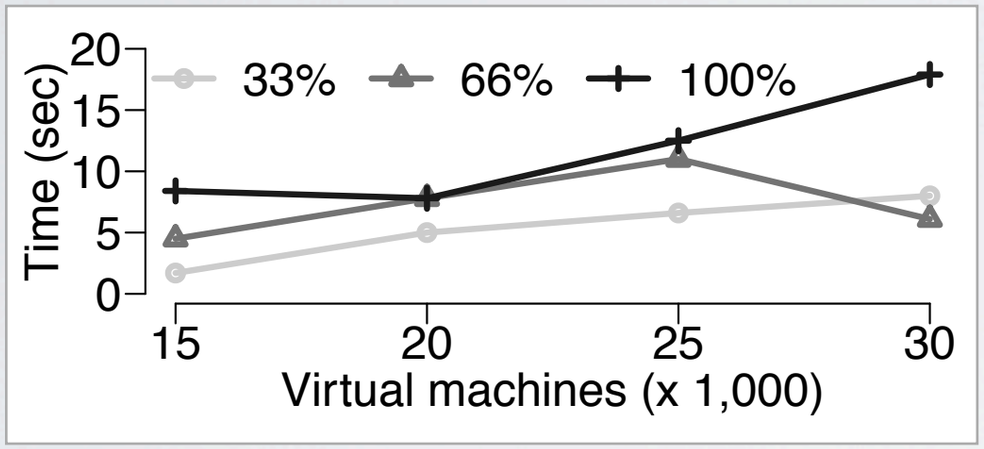


Reconfiguration duration

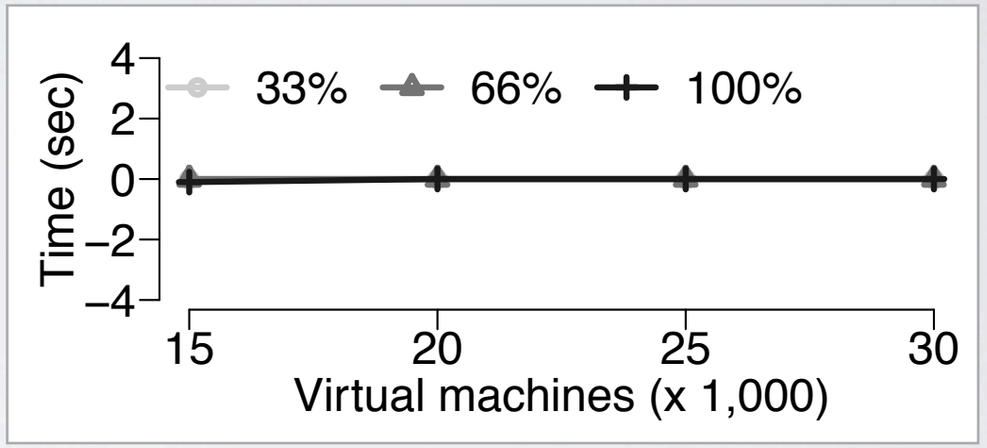
- reduces the solving duration
- reduces the delay to start actions

THE PLACEMENT CONSTRAINTS

NR case



Solving duration

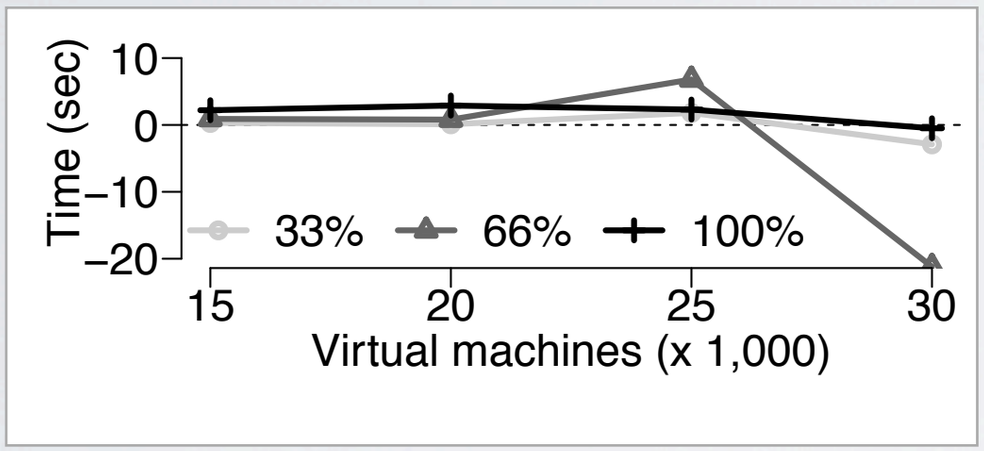


Reconfiguration duration

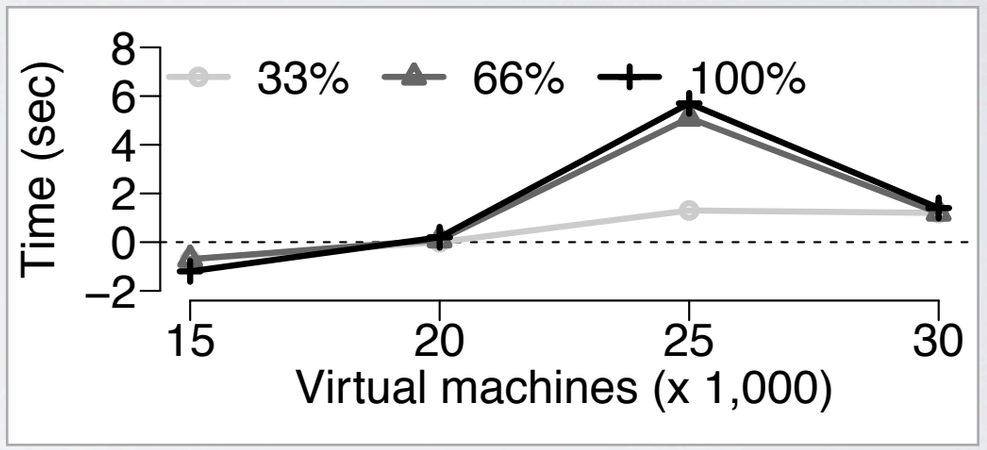
- the core-RP resolution dominates the solving duration
- no impact on the reconfiguration plans

THE PLACEMENT CONSTRAINTS

LI case



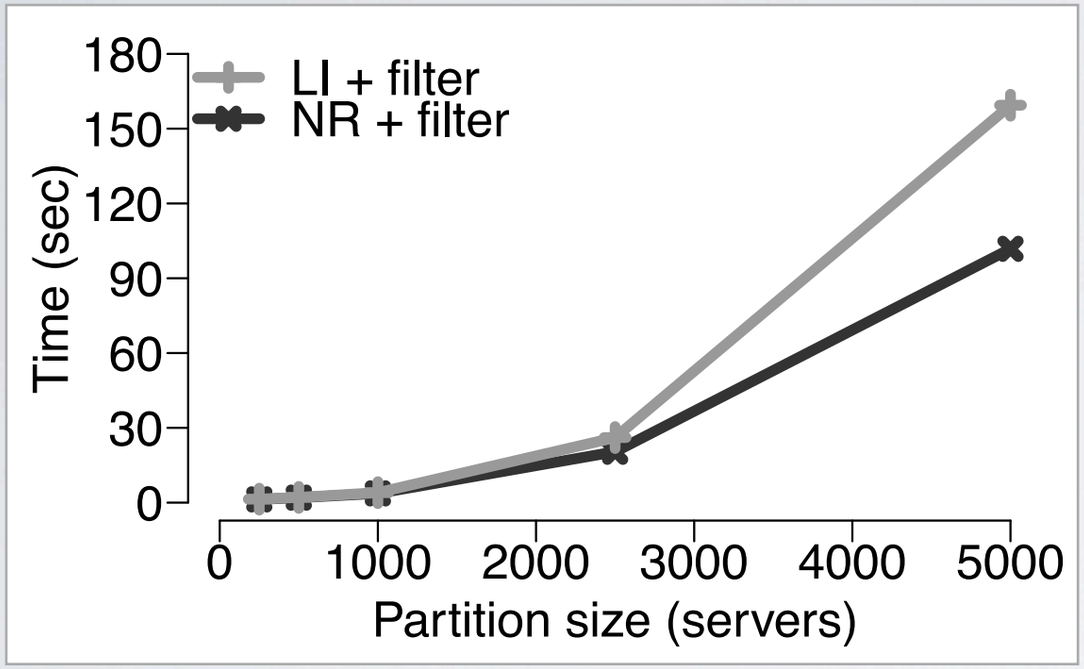
Solving duration



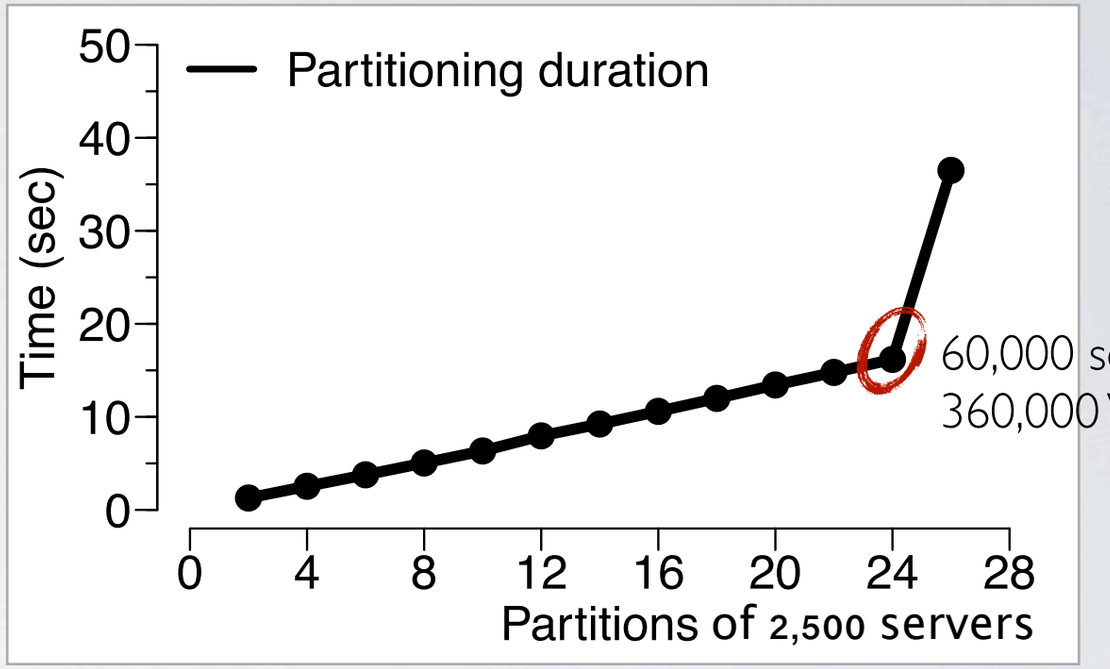
Reconfiguration duration

- no or negative overhead
- placement constraints simplifies the core-RP resolution
- except during the phase transition, no impact on the plans

PARTITIONING



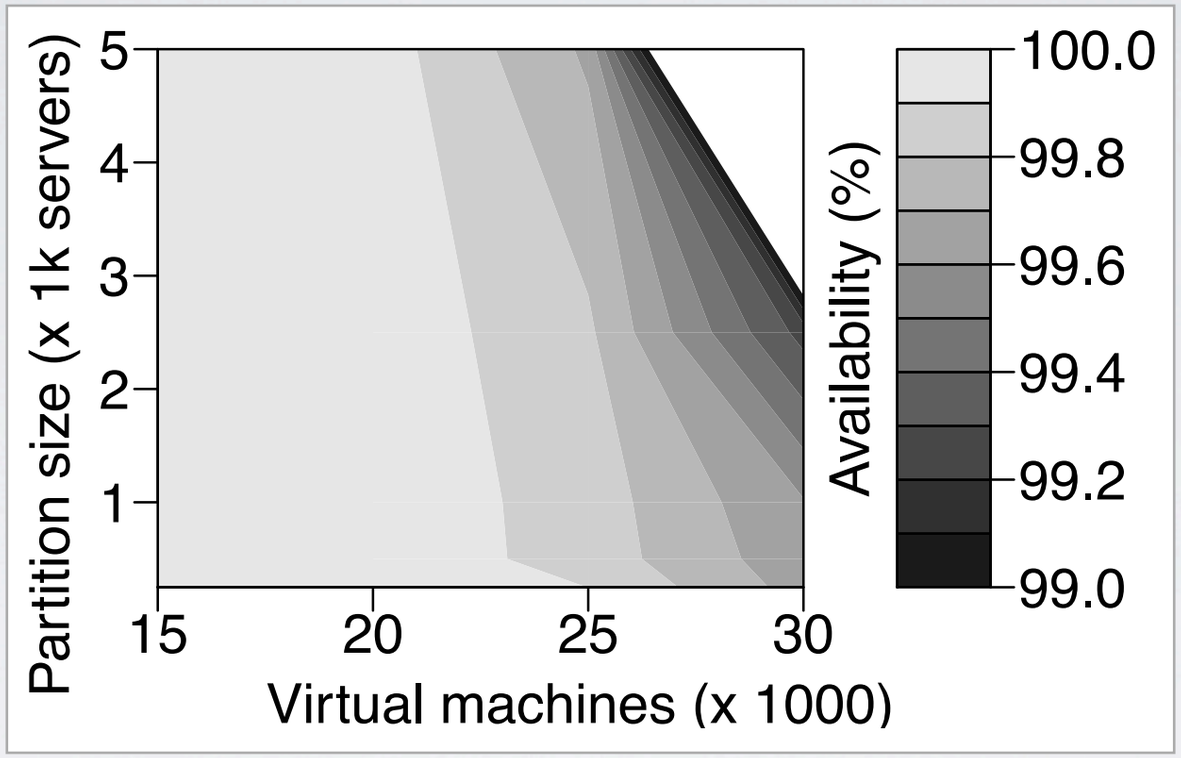
Solving duration



Partitioning duration

- reduces the solving duration
- the number of slaves to solve sub-RPs limits the scalability
- no impact on the quality of the reconfiguration plans
- too small partitions may alter the solvability

GLOBAL AVAILABILITY



The operator can establish a trade-off between:

- a high resource usage (big consolidation ratio)
- resource fragmentation (partitions size)

BTRPLACE

- a VM placement algorithm extensible by design
- declarative configuration scripts to state the constraints
- expressivity : constraints cover several concerns
- scalability through partitioning
- part of the open source OW2 - Entropy

The next BtrPlace

- new constraints, new concerns
- automatic, optimistic partitioning
- violatable constraints with context-aware penalties

ABOUT BTRPLACE

Online demo : http://btrp.inria.fr/sandbox

The Btrplace constraint catalog (draft):
http://www-sop.inria.fr/members/Fabien.Hermenier/btrpcc/

Publications on my webpage :

http://sites.google.com/site/hermenierfabien/

SOME PUBLICATIONS

The origins with Entropy

Entropy: a consolidation manager for cluster. F. Hermenier, X. Lorca, J.-M. Menaud, G. Muller, J. Lawall. In VEE 2009

Toward Btrplace through use cases:

Fault tolerance: Dynamic Consolidation of Highly-Available Web Applications. F. Hermenier, J. Lawall, J.-M. Menaud, G. Muller. Research Report 2011

An energy aware framework for VMs placement in cloud federated data centres. C. Dupont, G. Giuliani, F. Hermenier, T. Shulze, A. Somov, E-energy 2012

The theory behind Btrplace:

Bin Repacking Scheduling in Virtualized Datacenters. F. Hermenier, S. Demassey, X. Lorca. In CP 2011

The no-longer cursed paper about Btrplace fundaments (this talk):

Btrplace: A Flexible Consolidation Manager for Highly Available Applications. F. Hermenier, J. Lawall, G. Muller. *To appear* in IEEE TDSC 2013