# BIN REPACKING SCHEDULING IN VIRTUALIZED DATACENTERS

Fabien HERMENIER, FLUX, U. Utah & OASIS, INRIA/CNRS, U. Nice
Sophie DEMASSEY, TASC, INRIA/CNRS, Mines Nantes
Xavier LORCA, TASC, INRIA/CNRS, Mines Nantes

# CONTEXT

## datacenters and virtualization

# DATACENTERS

interconnected **servers**

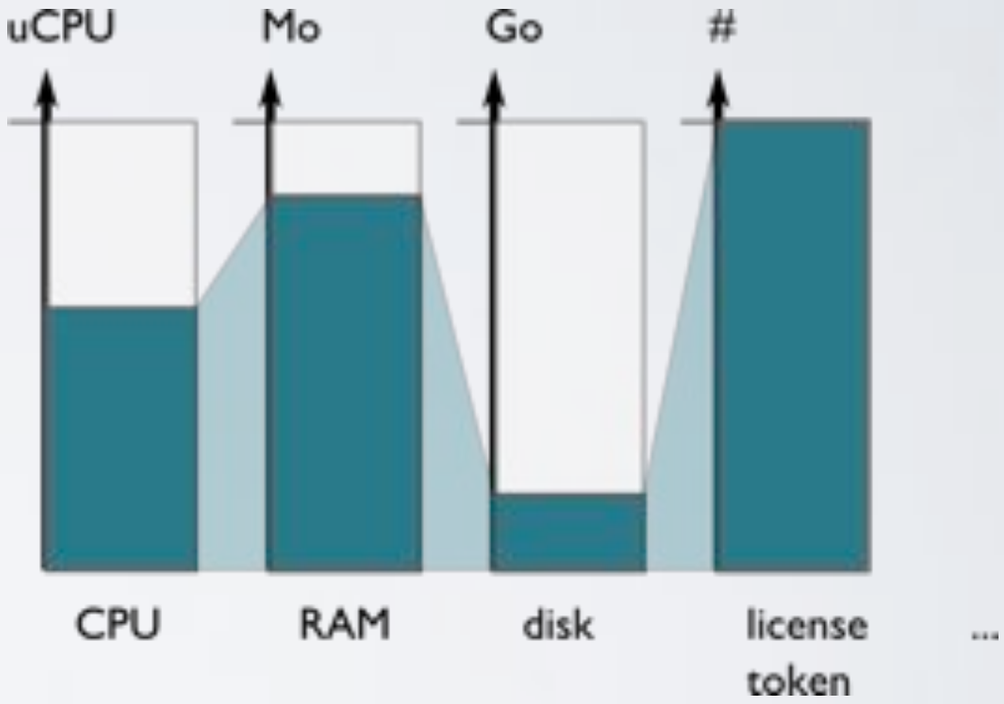
hosting

distributed **applications**

# RESOURCES

server
**capacities**

application
**requirements**

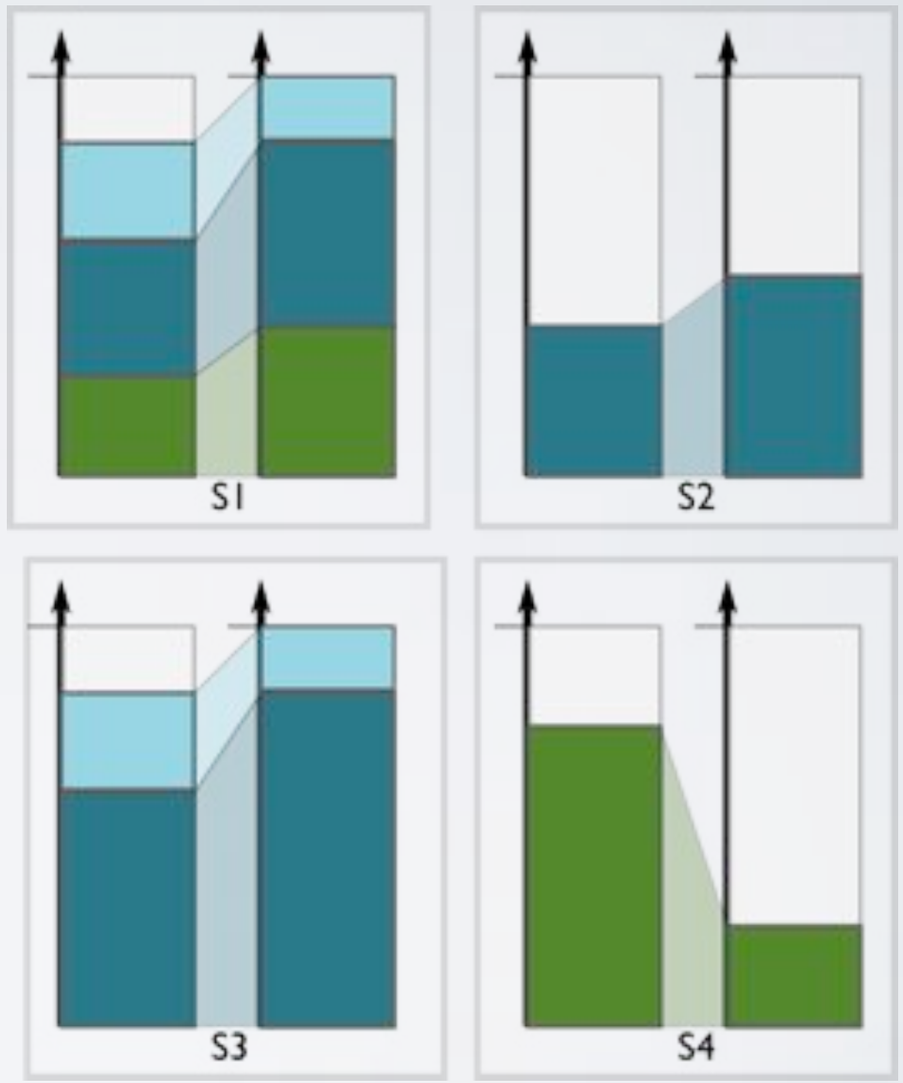


# VIRTUALIZATION

applications embedded in
**Virtual Machines**

**colocated** on any server

**manipulable**




1 datacenter
4 servers, 3 apps

# ACTION

- stop/suspend
- launch/resume
- migrate live

has a known **duration**

**consumes** resources

**impacts** VM performance

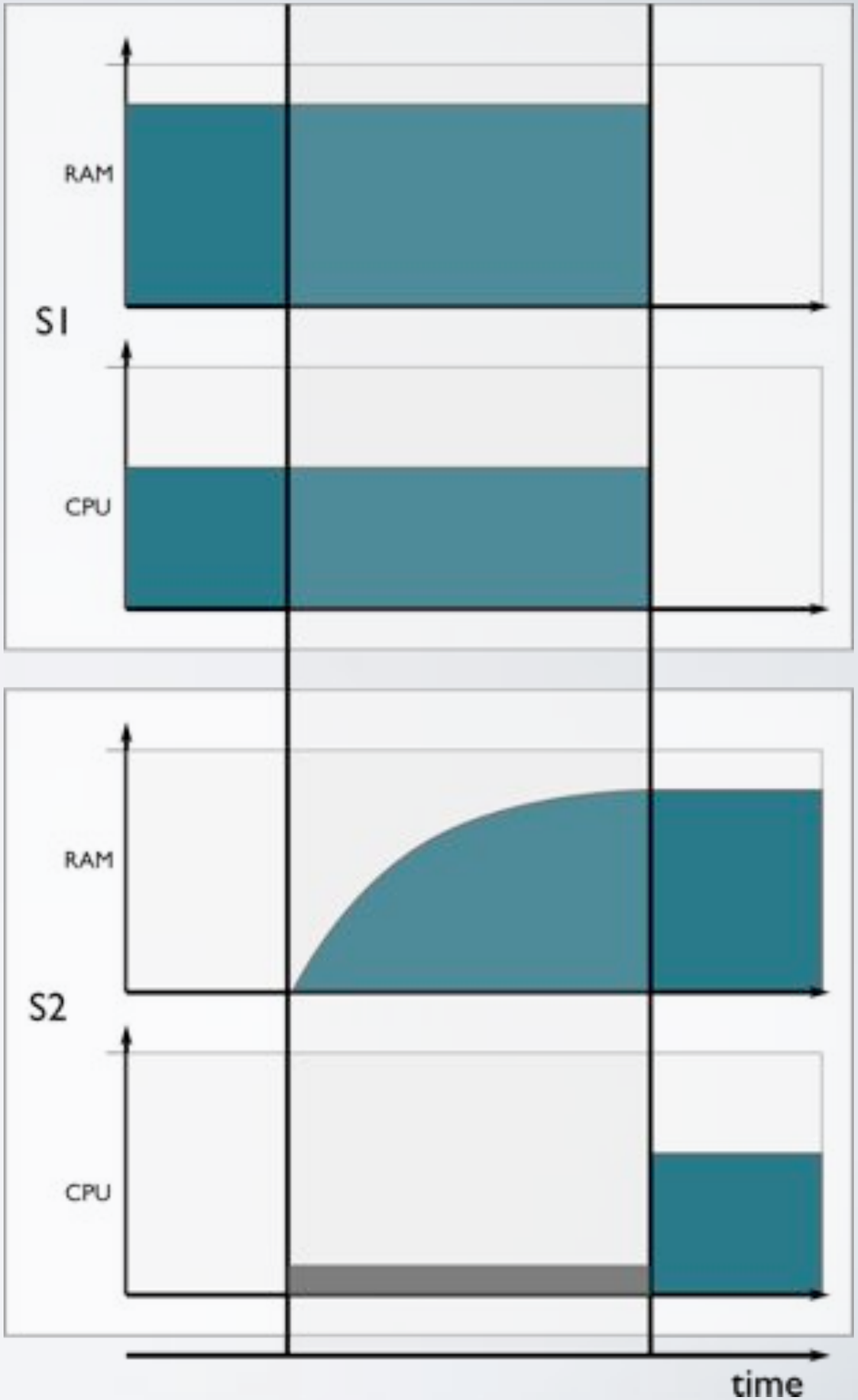
# ACTION

live migration

has a known **duration**

**consumes** resources

**impacts** VM performance



# DYNAMIC SYSTEM

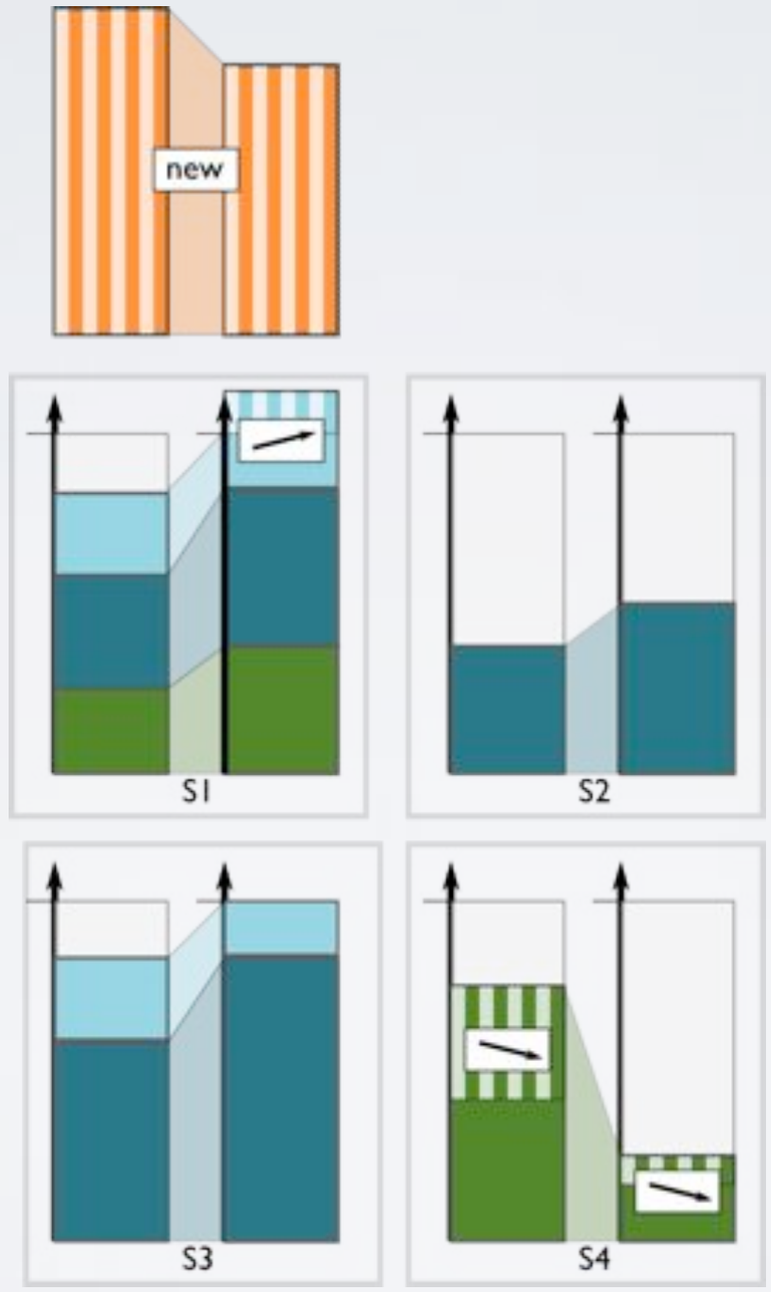
## Applications

- submission
- removal (complete, crash)
- requirement change (load spikes)

## Servers

- addition
- removal (power off, crash)
- availability change

# DYNAMIC RECONFIGURATION



# RECONFIGURATION PLAN

# RECONFIGURATION PROBLEM

given an initial configuration and new requirements:

- find a new viable configuration
- associate actions to VMs
- schedule the actions to resolve violations and dependencies

s.t every action is complete as early as possible

# + USER REQUIREMENTS

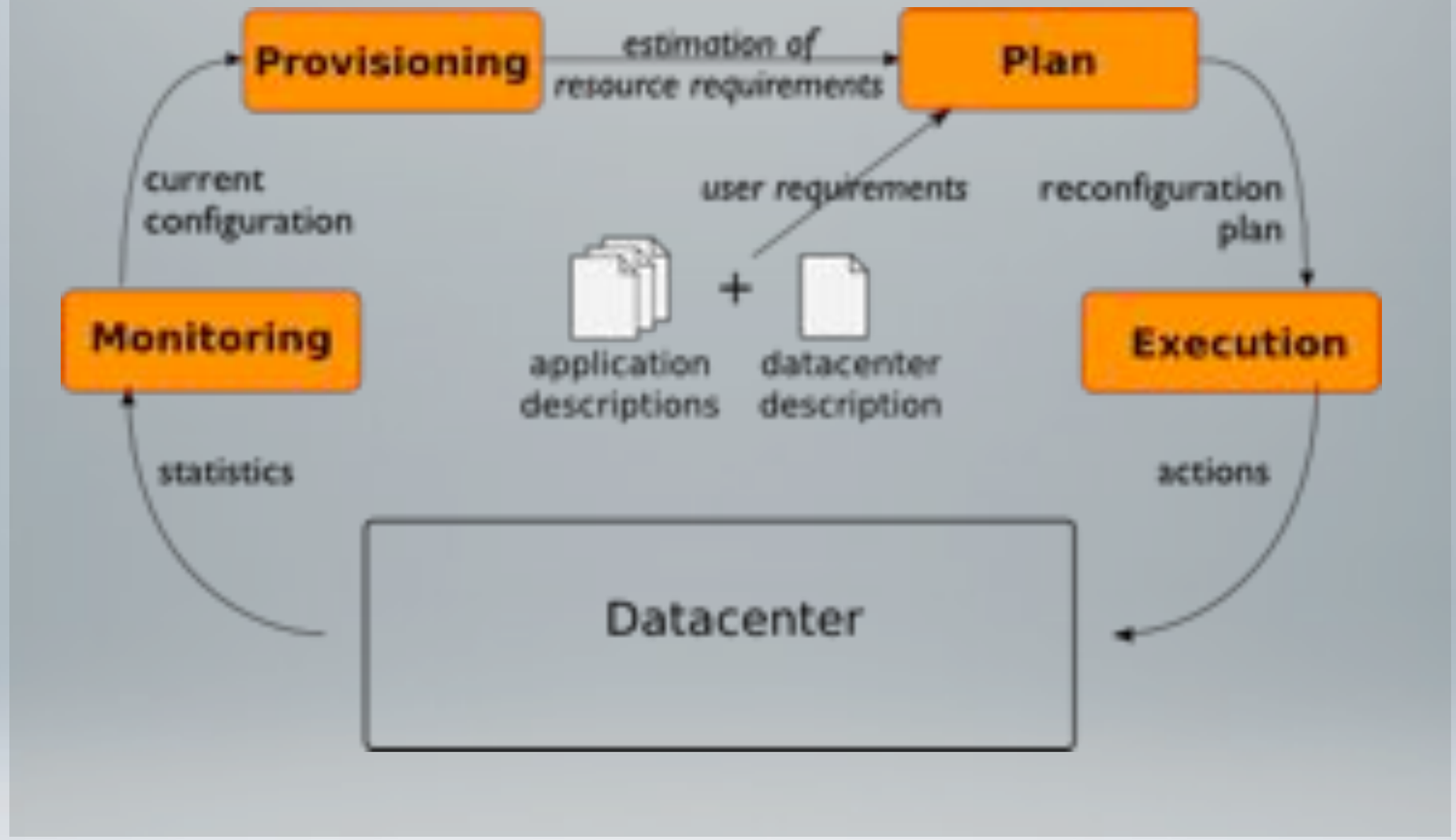
**Clients**

- fault-tolerance w. replication
- performance w. isolation
- resource matchmaking

**Administrators**

- maintenance
- security w. isolation
- shared resource control

to satisfy at any time **during** the reconfiguration




# ENTROPY

## an autonomous VM manager

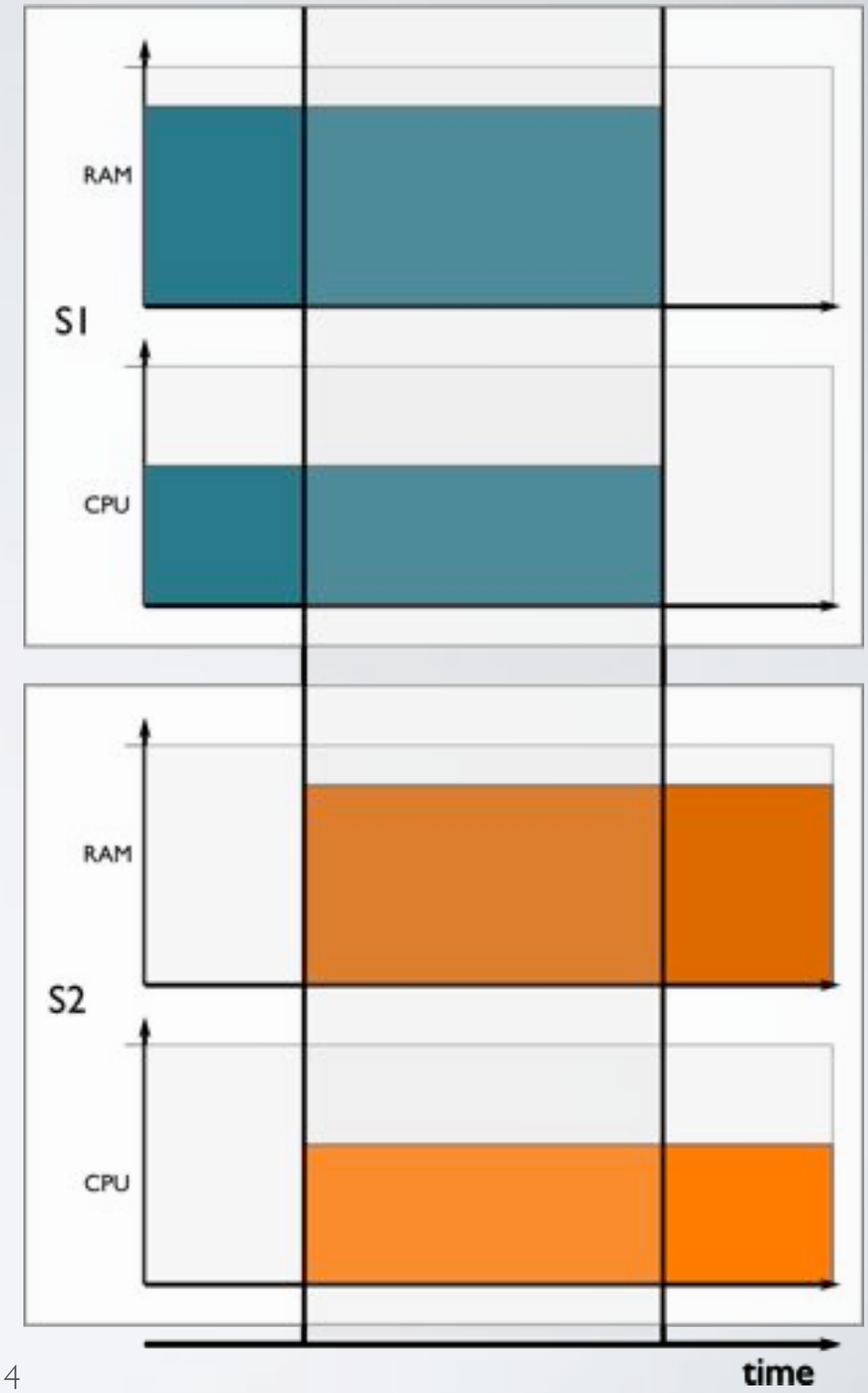
# PLAN MODULE

- reconfiguration problem: vector packing + cumulative scheduling + side constraints (NP-hard)
- trade-off fast+good
- composable online
- easily adaptable offline

Constraint Programming

# ABSTRACTION

- 1VM = **0 or 1 action**
- **min** $\sum_{\text{actions A}}$ end(A)
- **full** resource requirements at transition



# Compound CP model

# Packing + Scheduling

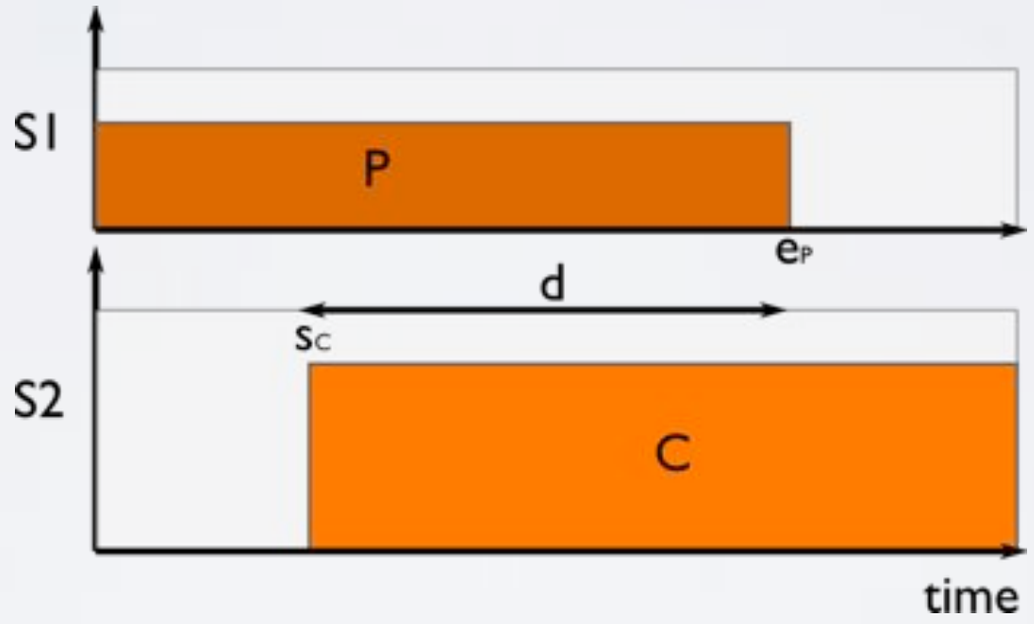
**decomposition degrades** the objective and may result in a feasible packing without reconfiguration plan

**shared constraints** resource+side

**separated branching heuristic** 1-packing 2-scheduling

# Producer/Consumer

1 action = **2 tasks** / **no-wait**

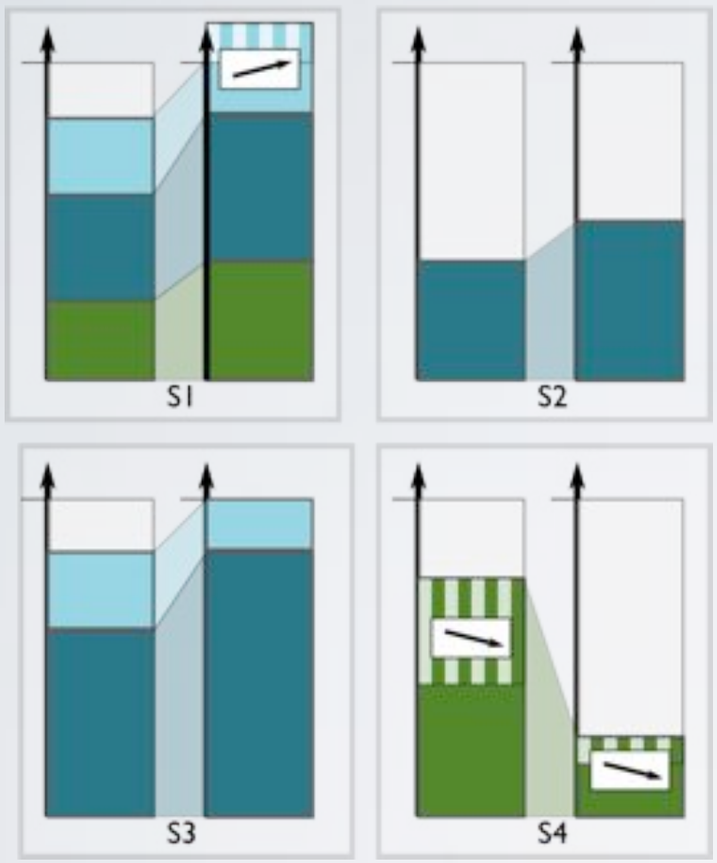



home-made `cumulatives` in every dimension

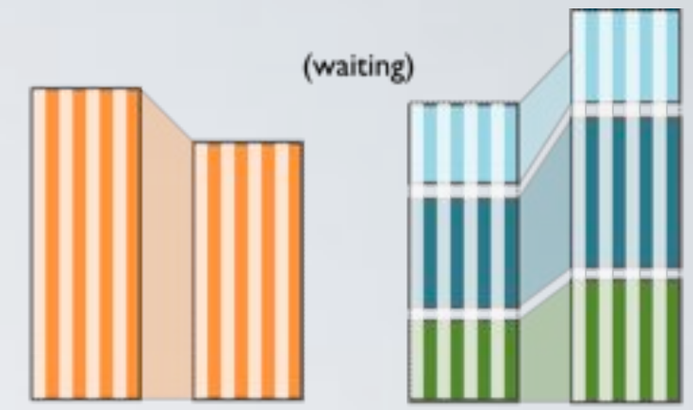
# SIDE CONSTRAINTS

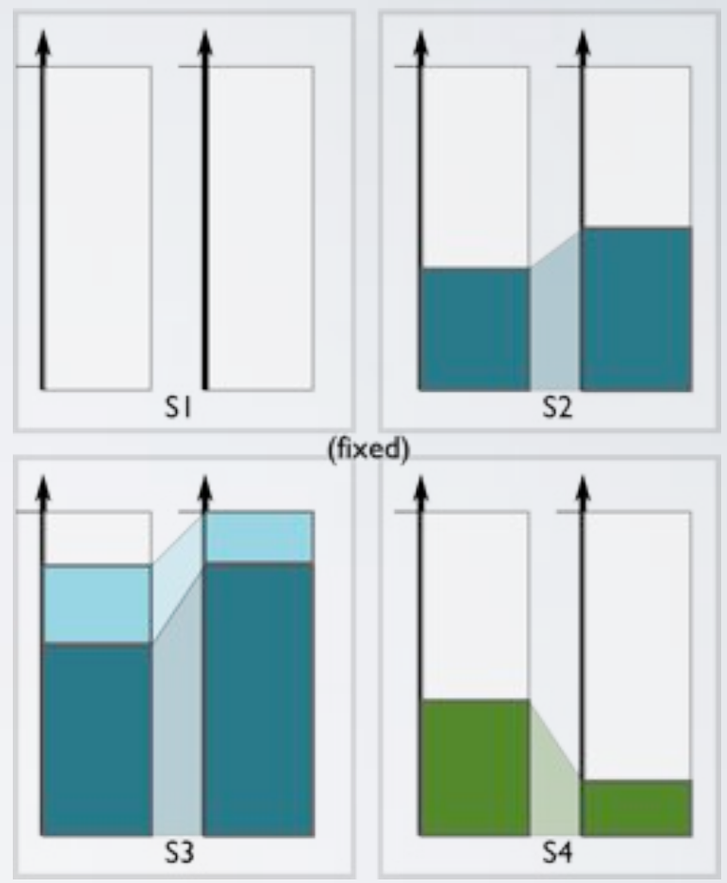
| | |
|---|---|
| *ban* | unary |
| *fence* | unary |
| *spread* | `alldifferent` + precedences |
| *lonely* | `disjoint` |
| *capacity* | `gcc` |
| *among* | `element` |
| *gather* | `allequals` |
| *mostly spread* | `nvalue` |

# REPAIR


candidate VMs
fixed
heuristically




resource+placement constraints come with a **new service**:
**return a feasible sub-configuration**

# EVALUATION

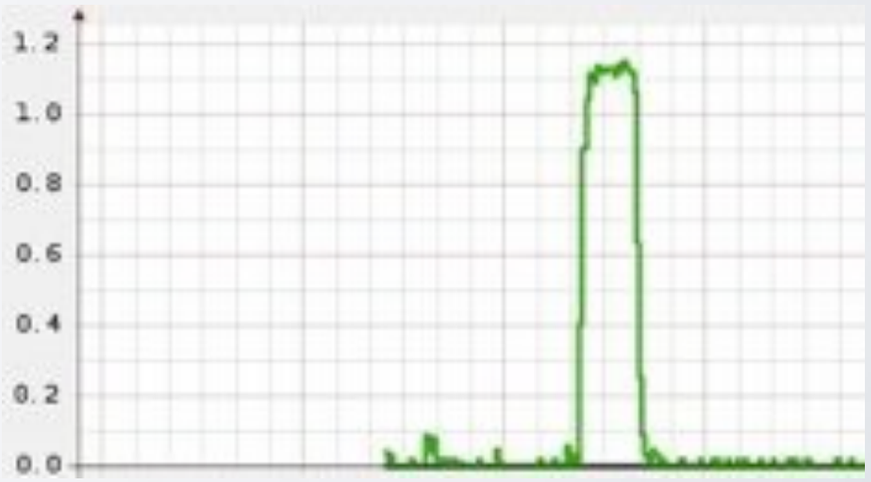
# PROTOCOL

500 - 2,500 homogeneous servers

amazon web services™

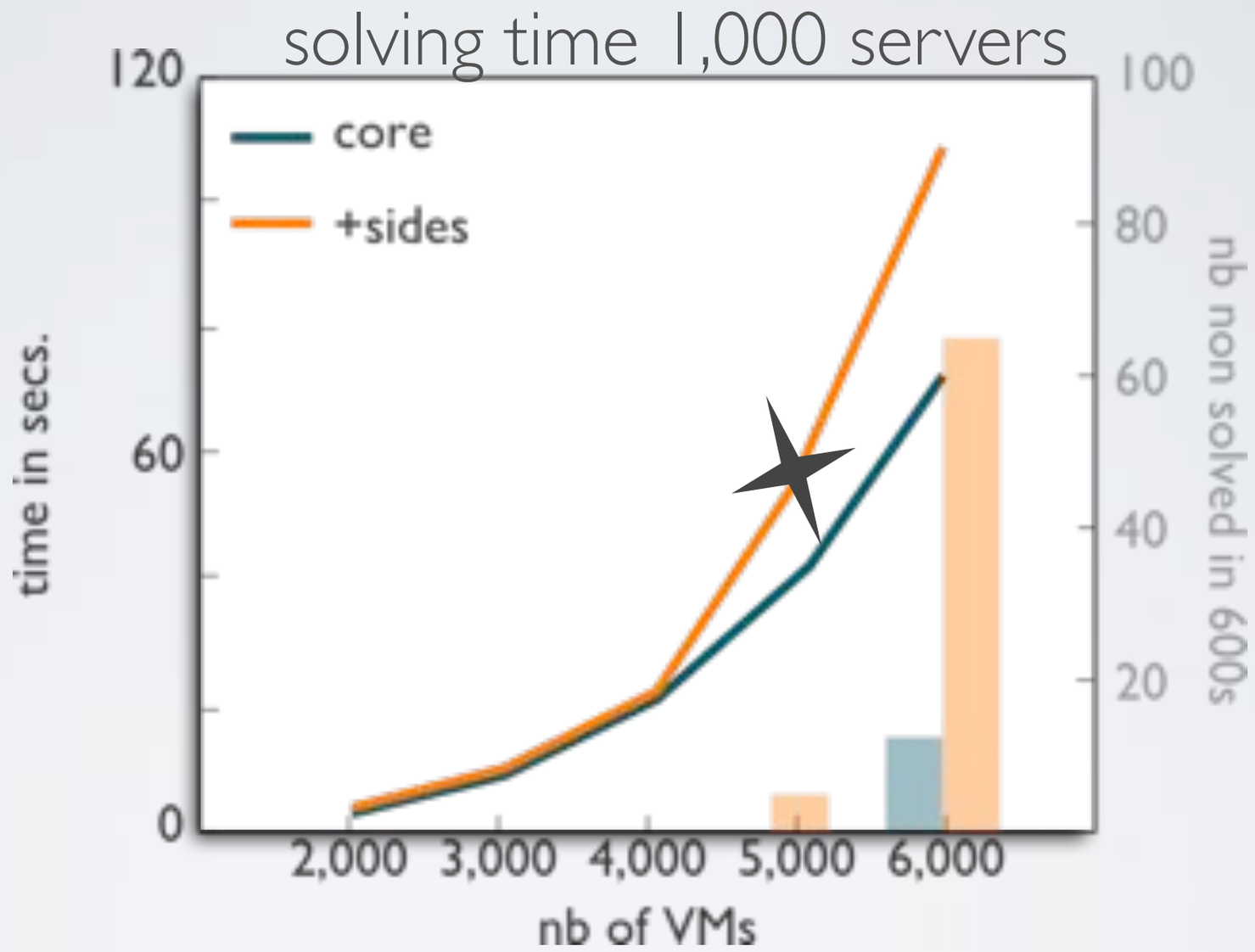
2,000 - 10,000 heterogeneous VMs

extra-large/high-memory EC2 instances
3-tiers HA web applications with replica



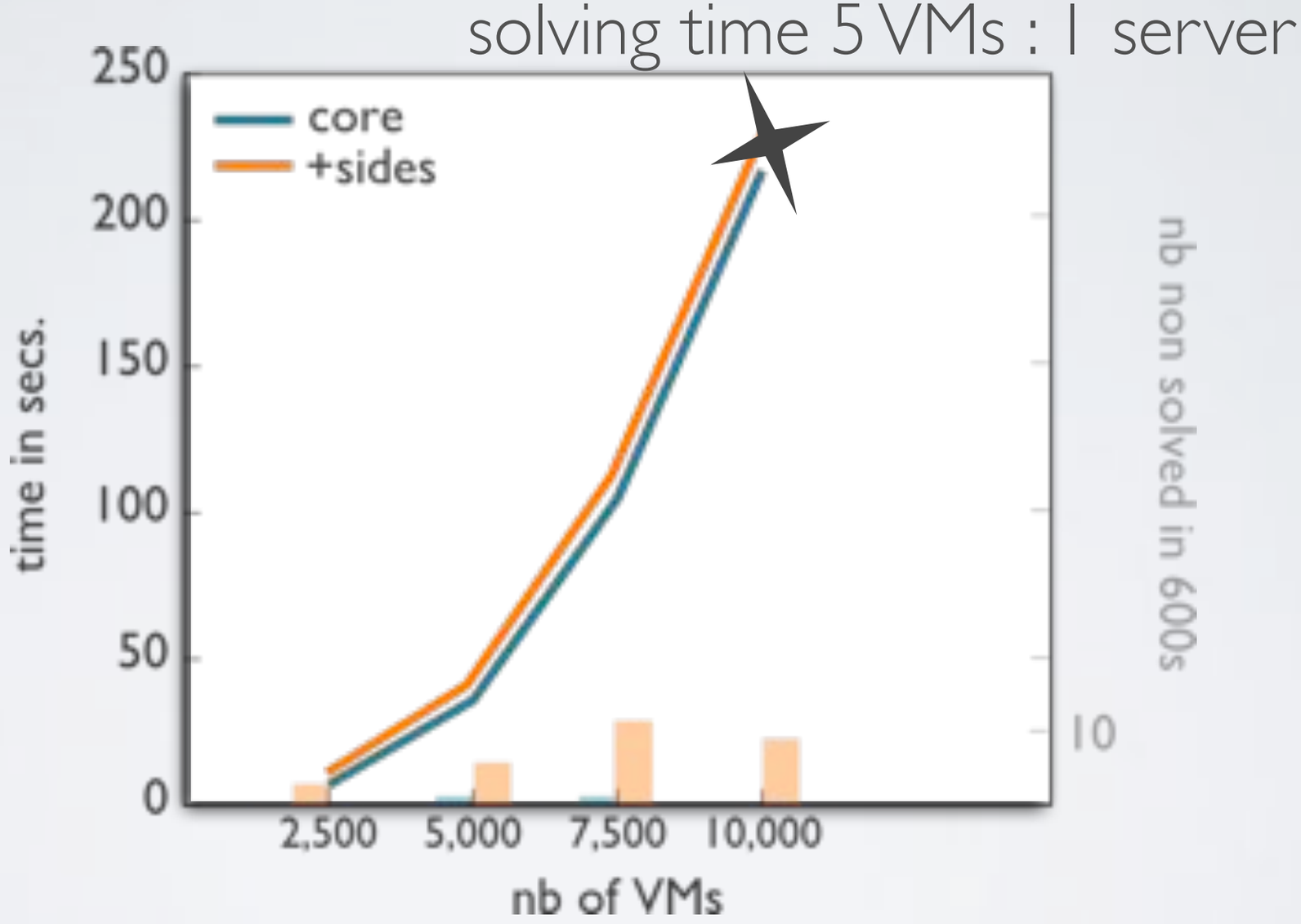

50% VM grow 30% uCPU
4% VM launch, 2% VM stop
1% server stop

# LOAD




70% = standard average consolidation ratio

# SCALABILITY




2,000 servers = standard capacity of a container

field to investigate for packing+**scheduling**

≠ **usages** to optimize: CPU, energy, revenue

**side constraints** important for users

**Entropy: CP**-based resource manager, fast, scalable, generic, composable, flexible, easy to maintain

http://entropy.gforge.inria.fr/

# CONCLUSION

user constraint **catalog**

routing and application **topology** constraints

**soft/explained** user constraints

new results available:

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

# PERSPECTIVES