BTRPLACE

Facing SLA expectations in a cloud

Université Nice SOPHIA ANTIPOLIS

Fabien Hermenier

fabien.hermenier@unice.fr

ECOLE DES MINES DE NANTES

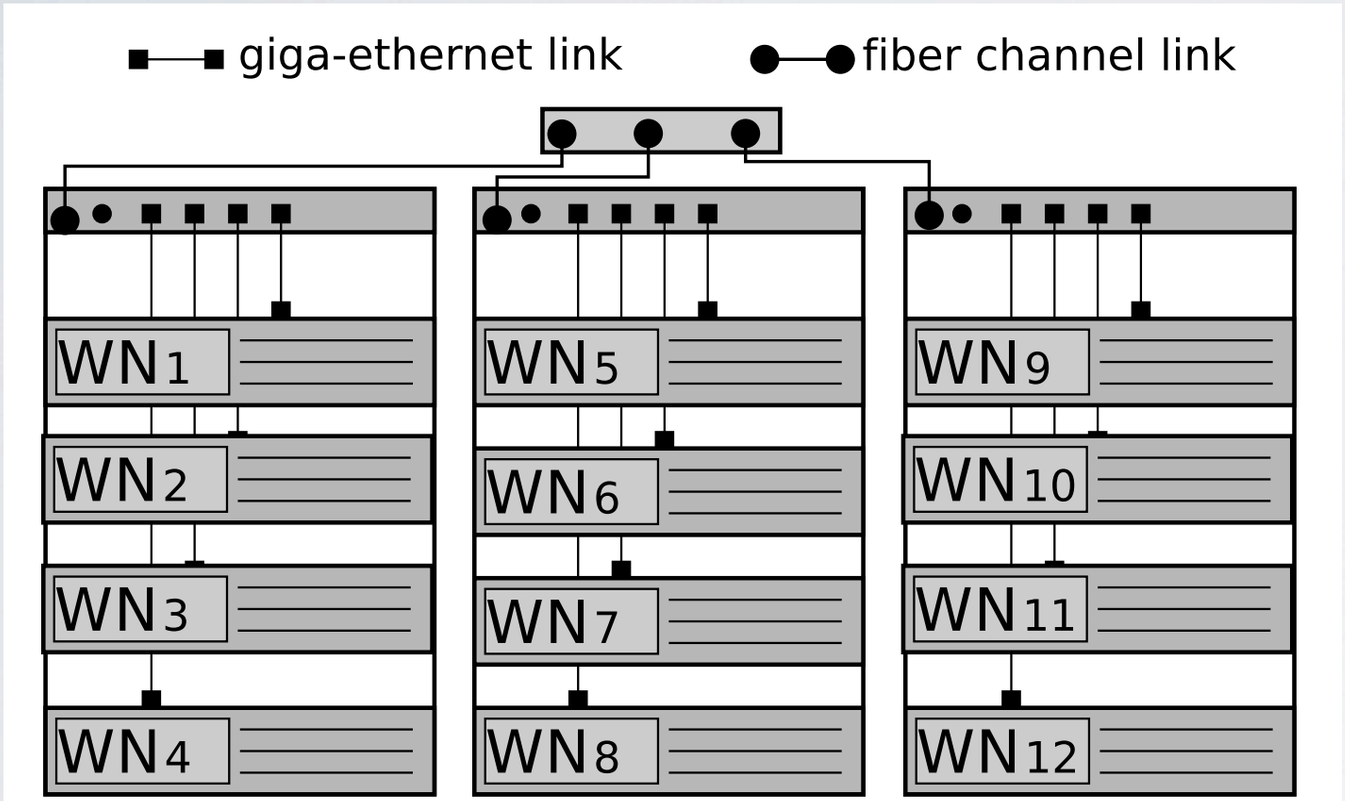
Jean-Marc Menaud

menaud@mines-nantes.fr

OW2con'12

OW2Con @ Orange labs 27-29 nov. 2012

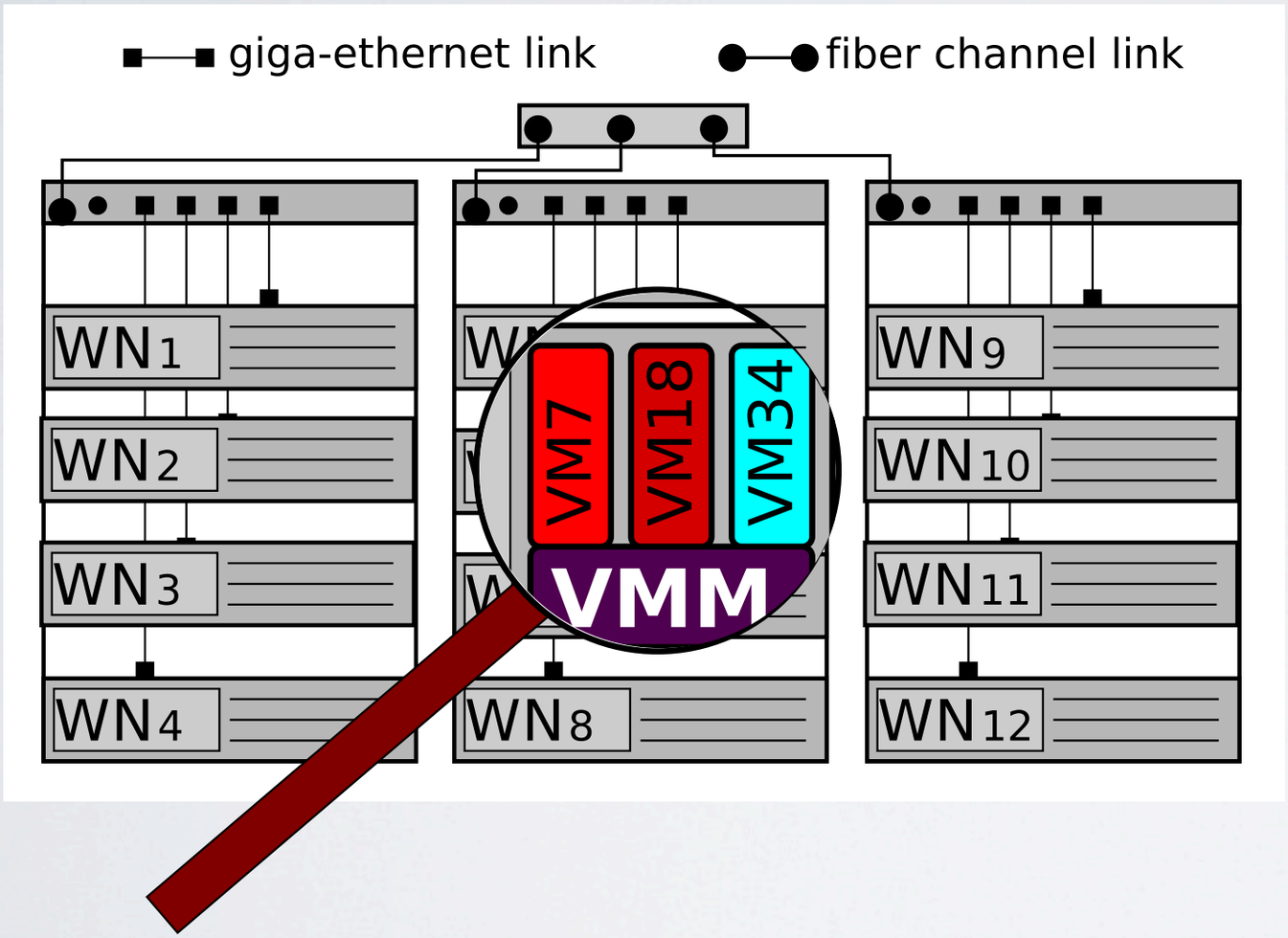
HOSTING PLATFORMS



Sysadmins are looking for:

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

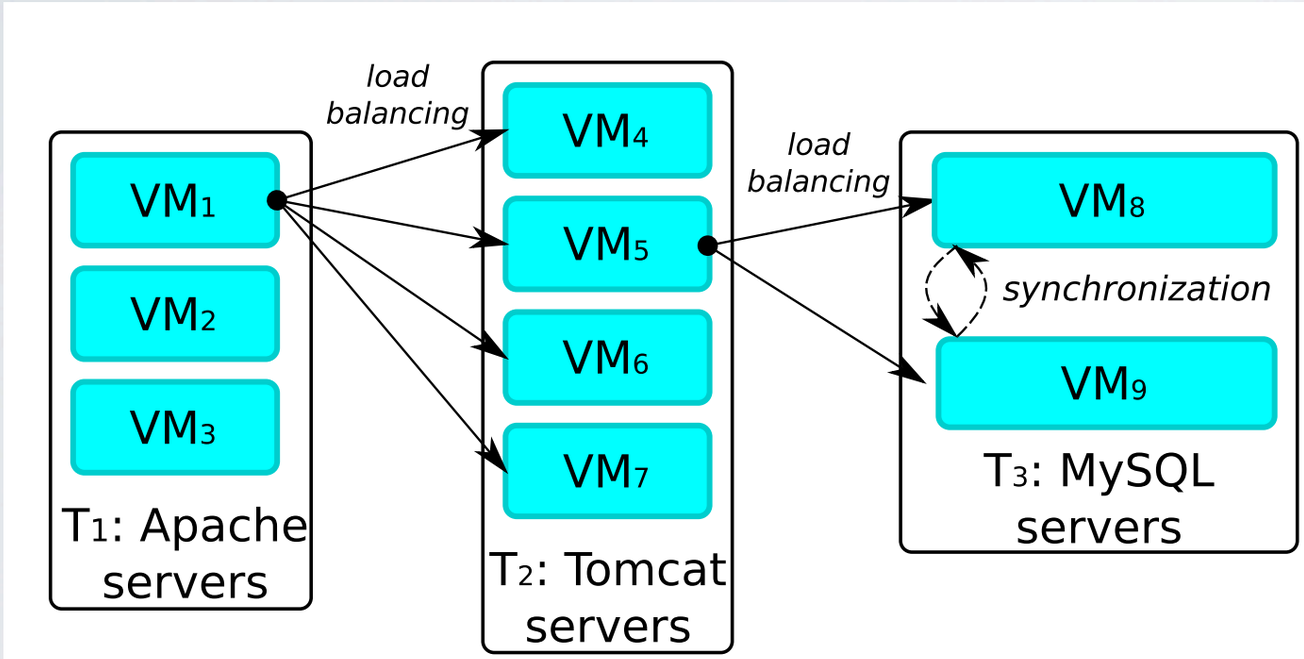
HOSTING PLATFORMS



Sysadmins are looking for:

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

VIRTUAL APPLIANCE



Clients are looking for:

- performance
- reliability
- isolation
- ...

PLACEMENT CONSTRAINTS

An unachieved story in which you are not the hero

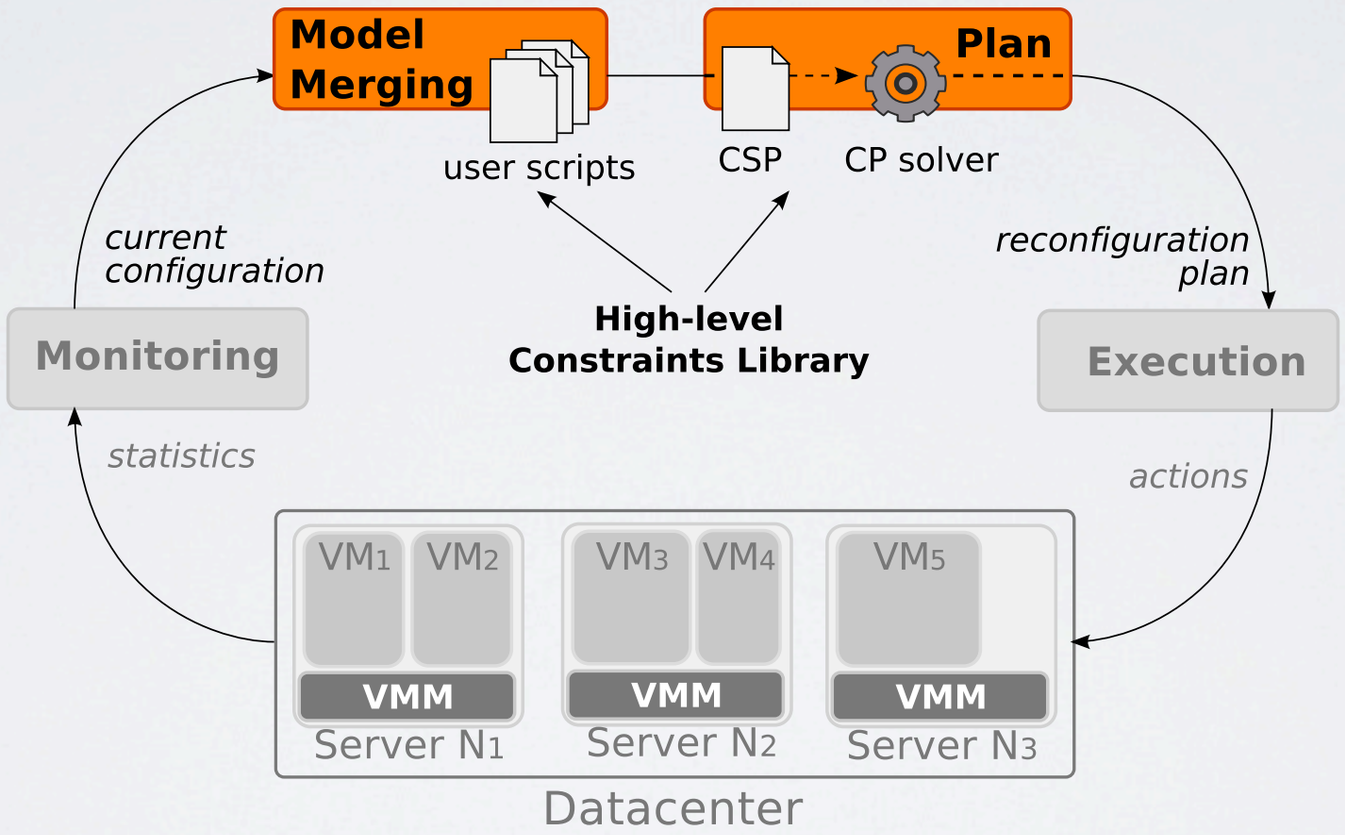
- closed-source algorithms
- not extensible algorithms by design

PLACEMENT CONSTRAINTS

- you have peculiar expectations
- you should be able to tune your placement algorithms
- make your needs our researches

BTRPLACE

From a Entropy built-in to standalone VM placement algorithm



✓ flexibility

✓ composability

BTRPLACE

Placement constraints:

fault tolerance `splitAmong, spread` isolation `split, lonely, quarantine` infrastructure management `cumulatedCapacity, fence, root, ban, singleCapacity, online, offline, running, sleeping, terminate, among` performance `cpuMargin ,gather, preserve, oversubscription` energy management `maxOnline, noIdlesOnline, minSpareResources, maxSpareResources, ...`

Optimization objectives:

`«fast reconfigurations», «load balancing», «low energy consumption», «low gas emissions», ...`

THEY TRUST BTRPLACE

AGENCE NATIONALE DE LA RECHERCHE ANR

SelfXL
Self-management of compleX and Large scale systems

EasyVirt

open Cloudware

FIT 4 GREEN

orange™

Btrcloud

OW2 Sirocco-vmm

OW2con'12

BTRPLACE

- an extensible, composable VM placement algorithm
- a part of the OW2 - Entropy
- open source LGPL V3 Free Software
- a research project since 2006
- 10 publications, 2 awards
- academic and industrial partners
- contacts: `fabien.hermenier@unice.fr`
  `menaud@mines-nantes.fr`

Try it: http://btrp.inria.fr/sandbox

PROGRAMMING PLACEMENT CONSTRAINTS

express the placement you want :

```
//LazySpread: future running VMS must run on distinct nodes
List<IntDomainVar> runnings = new ArrayList<IntDomainVar>();

for (VirtualMachine vm : getAllVirtualMachines()) {
  if (core.getFutureRunnings().contains(vm)) {
    Slice t = core.getAssociatedAction(vm).getDemandingSlice();
    if (t != null) {runnings.add(t.hoster());}
  }
}
core.post(new BoundAllDiff(runnings.toArray(), true));
```

let Constraint Programming solve that for you !