BTRPLACE
Facing SLA expectations in a cloud

Fabien Hermenier
fabien.hermenier@unice.fr

Jean-Marc Menaud
menaud@mines-nantes.fr
HOSTING PLATFORMS

Sysadmins are looking for:
- manageability
- security
- efficient resource usage
- ...

- giga-ethernet link
- fiber channel link
HOSTING PLATFORMS

Sysadmins are looking for:
• manageability
• security
• efficient resource usage
• ...
VIRTUAL APPLIANCE

Clients are looking for:
• performance
• reliability
• isolation
• ...
PLACEMENT CONSTRAINTS

- VM-host affinity (DRS 4.1) - apr. 2011
- Dedicated instances (EC2) - mar. 2012
- MaxVMsPerServer (DRS 5.1) - sep. 2012

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

Btrcloud

ow2 Sirocco-vmm
BTRPLACE

• an extensible, composable VM placement algorithm
• a part of the OW2 - Entropy
• open source LGPLv3
• 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:

```java
//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!