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

Fabien Hermenier
fabien.hermenier@unice.fr

Research Team OASIS, INRIA/I3S
HOSTING PLATFORMS

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

[Diagram showing giga-ethernet and fiber channel links with WN1, WN2, WN3, WN4, WN5, WN6, WN7, WN8, WN9, WN10, WN11, WN12]
HOSTING PLATFORMS

Operators are looking for:
- manageability
- security
- efficient resource usage
- ...
VIRTUAL APPLIANCE

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

T1: Apache servers
VM1
VM2
VM3

T2: Tomcat servers
VM4
VM5
VM6
VM7

T3: MySQL servers
VM8
VM9

load balancing

load balancing

synchronization
PLACEMENT CONSTRAINTS

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

### Diagram

- **VM-host affinity (DRS 4.1)**: Apr. 2011
- **Dedicated instances (EC2)**: Mar. 2012
- **MaxVMsPerServer (DRS 5.1)**: Sep. 2012
- **The constraint I needed in 2012**: ?? 2013
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
• provide datacenter and appliances descriptions
• human friendly definition of a viable datacenter
SAMPLE RECONFIGURATION

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)

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

*Diagram showing the placement and scheduling of d-slices and c-slices.*
MODELING THE PLACEMENT CONSTRAINTS

Using variables of the core-RP:

- \( c_{\text{host}} \): Current host of the VM (constant)
- \( c_{\text{men}}, c_{\text{cpu}} \): Current amount of memory and uCPU resources allocated to the VM (constant)
- \( c_{\text{ed}} \): Time the VM may leave its current host
- \( d_{\text{host}} \): Next host of the VM
- \( d_{\text{men}}, d_{\text{cpu}} \): Next amount of memory and uCPU resources to allocate to the VM
- \( d_{\text{st}} \): Time the VM arrives on its next host

**spread\(\{\text{VM1}, \text{VM2}\}\):**

\[
\text{allDifferent}(d_{\text{host}}^1, d_{\text{host}}^2) \land \\
\begin{align*}
  d_{\text{host}}^1 &= c_{\text{host}}^2 \rightarrow d_{\text{st}}^1 \geq c_{\text{ed}}^2 \land \\
  d_{\text{host}}^2 &= c_{\text{host}}^1 \rightarrow d_{\text{st}}^2 \geq c_{\text{ed}}^1
\end{align*}
\]
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

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

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]);
OPTIMIZING THROUGH PARTITIONING

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

• is Brplace flexible in practice?

• does Btrplace make 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

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
<th>Reconfiguration Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td>2'10</td>
<td>+ban({WN8})</td>
<td>3 + 3 relocations in 0'42</td>
</tr>
<tr>
<td>4'30</td>
<td>+ban({WN4})</td>
<td>2 + 7 relocations in 1'02</td>
</tr>
<tr>
<td>7'05</td>
<td>-ban({WN4})</td>
<td>no reconfiguration</td>
</tr>
<tr>
<td>11'23</td>
<td>+ban({WN4})</td>
<td>no solution</td>
</tr>
<tr>
<td>11'43</td>
<td>-ban({WN8})</td>
<td></td>
</tr>
<tr>
<td></td>
<td>+ban({WN4})</td>
<td>2 relocations in 0’28</td>
</tr>
</tbody>
</table>

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

- reduces the solving duration
- reduces the delay to start actions
THE PLACEMENT CONSTRAINTS

NR case

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

Li case

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

- 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
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


Toward Btrplace through use cases:


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:


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