Virtualization, resource management and autonomous systems

Gastón Keller
PhD Candidate

The University of Western Ontario
Overview

- Virtualization
- Uses of virtualization
- Virtualization in data centers
- VM replication
- VM memory management
Overview

- **Virtualization**
- Uses of virtualization
- Virtualization in data centers
- VM replication
- VM memory management
1960s: IBM designed the operating system CP-40/CMS.

Provided VMs that were indistinguishable from real machines by user programs.
Role in Industry

Level of adoption around 20 to 30% (expected to grow another 20%). *

Principal motivations:
- cost-cutting
- business continuity
- server manageability

Taxonomy of VMs

Source: Smith, J. E.; Nair, R.; *The Architectures of Virtual Machines.*
Definition

"(System) Virtualization is a software technique that enables the simultaneous execution of multiple computer systems in one physical machine."
Hypervisor-based Virtualization

Type 1 Hypervisor

Type 2 Hypervisor
Interprets the code of the guest OS and its applications.
Type 1 Hypervisor

It can interpret the code of the guest OS as a Type 2 Hypervisor or use paravirtualization.
Paravirtualization

The guest OS source code is modified to enable communication with the hypervisor.
In 2006, Intel and AMD released CPUs with support for virtualization.

**Full-virtualization** enables unmodified guest OSes to run in VMs.

However, *paravirtualization* still offers better performance.
Evolution of paravirtualization

* Check out paravirt_ops in Linux.
Xen

Started as a research project at the University of Cambridge (2003).

Is Open Source Software (which contributed to its success).

Supported by big companies such as IBM, Intel, and Oracle (among others).
Resource Management

(CPU) Credit Scheduler:
- weight value
- cap value

Memory:
- reservation value
- maximum value
- minimum value (Dom0)
Resource Management...

Networking:
- bridging
- routing

To reduce CPU overhead:
- dedicated NICs
- optimized inter-domain communication channel \((Dom0-DomU)\)
- virtualization aware NICs
OS-level Virtualization

Management SW
Privileged Container
Applications
Container
Applications

OS kernel
Hardware

USER SPACE

OpenVZ
server virtualization

VServer

Solaris
OpenVZ

Provides *operating system-level virtualization*.

Modified Linux to run multiple, isolated *containers*.

Basis of Parallels Virtuozzo Containers.
OpenVZ

Advantages:
• simple deployment
• close to native performance
• great scalability

Disadvantage:
• only GNU/Linux-based virtual environments
Resource Management

Organized in two levels:
  • storage subsystem
  • CPU scheduler
  • I/O scheduler

Memory:
  • *User Beancounters* (per container)
Overview

● Virtualization

● **Uses of virtualization**

● Virtualization in data centers

● VM replication

● VM memory management
Virtual Appliances

A software image containing a software stack (OS + app.) designed to run inside a virtual machine.

Makes software deployment easier and faster.

BitNami - http://bitnami.org/
VMware - http://www.vmware.com/appliances/
Potential benefits:

- resiliency
- scaling
- system-level portability
- observability

Palacios - http://www.v3vee.org/palacios/
Virtualization provides Grid Computing with *isolated, customized environments*.

In addition:
- legacy systems
- security
- flexible resource allocation
Grid Computing...

*Krsul et al.* developed **VMPlant** Grid service: flexible and efficient resource sharing through virtualization.

Components:

- **VMShop** (front-end)
- **VMPlants** (hosts)
Emeneker and Stanzione developed **Dynamic Virtual Clustering**: leverage an institution's clusters computing power through job forwarding and spanning.

VMs provided independence from the platform and encapsulation.
Overview

- Virtualization
- Uses of virtualization
- **Virtualization in data centers**
- VM replication
- VM memory management
Definition

"A data center is a collection of computing resources shared by multiple applications concurrently in return for payment by the application providers, on a per-usage basis, to the data center provider."
Data Centers

(*) http://scienceblogs.com/goodmath/2009/05/cloud_computing.php
Server Consolidation

(*) http://www.visualpharm.com/
Resource Stress Situations
Challenges

Virtualization brings benefits to the data center, but also challenges.

The research literature shows that unsolved issues are abound...
Challenges...

- Resource monitoring
- Algorithms and policies
- Resource management systems
- VM migration process
- Management tools
Wood et al. studied two approaches to monitoring:

- **black-box**, and
- **grey-box**.

Sandpiper used the data to detect **hotspots** and migrate VMs.
Algorithms and Policies

Gmach et al. studied workload consolidation through VM migration.

Developed:

- *placement controller*, and
  - (multiple policies)
- *migration controller*.
  - (multiple thresholds)
Zhu et al. developed a hierarchy of controllers:

- *node controller*,
- *pod controller*, and
- *pod set controller*.

Enable client and system admins to focus on policy setting.
Zhao and Figueiredo analyzed the VM migration process:

- in parallel,
- in sequence,
- cpu-intensive app.,
- mem-intensive app.

Predict time and performance of the VM migration process.
Vallée et al. extended OSCAR, a toolkit for cluster installation, configuration and management.

OSCAR-V enabled deployment and management of host OSes and VMs.
Challenges...

These were just a few research challenges that came with virtualization.

There are many more to be studied.
Overview

- Virtualization
- Uses of virtualization
- Virtualization in data centers
- VM replication
- VM memory management
Migration: moving a VM from one host node to another.

Replication: instantiating a copy of a VM in a different host node:
  - copy of current VM, or
  - instance of a stored image.
Our Work

- Built *Golondrina*, a resource management system for *OS-level virtualized* environments.

- Implemented *replication* mechanism to deal with resource stress situations.

- Compared *replication* mechanism with *migration* mechanism.
OS-level Virtualization
Our Work...

- Why use *migration* and *replication* to do resource management?

- Are both mechanisms needed? Why compared them?
Golondrina
Implementation

**OS:** CentOS 5.2 / OpenVZ

**Prog. Lang.:** Python

**Communications:** Twisted (event-driven networking engine)

**Load Balancer:** Pound
Basic Responsibilities

(C) Gather CPU statistics
(S) Process Clients' statistics
(S) Search for resource stress situations
(S) Determine sequence of relocations
(C) Execute migration/replication
Basic Responsibilities

(C) Gather CPU statistics
(S) Process Clients' statistics
(S) *Search for resource stress situations*
(S) *Determine sequence of relocations*
(C) *Execute migration/replication*
Resource Stress Check

- periodic check on (almost) every hardware node

\[
\text{overloaded} \leftarrow (\sum_{i=0}^{n} (\hat{u}_i > \text{threshold}) \geq k) \land (\hat{u}_{t+1} > \text{threshold})
\]

\(K\) out of the previous \(N\) checks were in excess

next predicted CPU utilization is in excess
Relocation Algorithm

1. decreasingLoadSort(stressed_HNs)
2. increasingLoadSort(non-stressed_HNs)
3. for each HN in stressed_HNs:
   4. decreasingLoadPolicy(containers)
   5. While HN is stressed:
      6. pick a CT and cycle through non-stressed_HNs until finding a HN_2
      8. that can host the CT
Replication Algorithm

1. generate CTID for the replica
2. bring CT image from central repository
3. process image
4. edit image configuration file
5. start replica
Experiments

• Cause resource stress situations (*httpperf* – load generator)

• Configure *Golondrina* to react:
  ➢ doing nothing
  ➢ using replication
  ➢ using migration

• Measure lost requests and throughput (*Apache web servers*)
Experiment 1

- 2 hardware nodes (*bravo02, bravo03*)
- 2 containers (*A, B*)
- *A* receives a load of around 70%
- *B* receives a load of around 105%
- *bravo02* experiences a load of 175%
## Results Exp. 1

<table>
<thead>
<tr>
<th>Web Server's Effectiveness</th>
<th>Nothing</th>
<th>Replication</th>
<th>Migration</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Servers</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>one.com</td>
<td>100.00%</td>
<td>99.11%</td>
<td>100.00%</td>
</tr>
<tr>
<td>two.com</td>
<td>100.00%</td>
<td>98.44%</td>
<td>100.00%</td>
</tr>
</tbody>
</table>

Throughput provided no conclusive results.
Experiment 2

- 2 hardware nodes \((bravo02, bravo03)\)
- 2 containers \((A, B)\)
- \(A\) and \(B\) receive a load of around 105%
- \(bravo02\) experiences a load of 200%
Results Exp. 2

<table>
<thead>
<tr>
<th>Web Server's Effectiveness</th>
<th>Nothing</th>
<th>Replication</th>
<th>Migration</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Servers</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>one.com</td>
<td>77.55%</td>
<td>87.55%</td>
<td>78.00%</td>
</tr>
<tr>
<td>two.com</td>
<td>62.44%</td>
<td>88.00%</td>
<td>84.44%</td>
</tr>
</tbody>
</table>

Throughput provided no conclusive results.
Experiment 3

- 2 hardware nodes (bravo02, bravo03)
- 4 containers (A, B, C, D)
- Each container receives a load of around 51%
- bravo02 experiences a load of 200%
## Results Exp. 3

<table>
<thead>
<tr>
<th>Servers</th>
<th>Nothing</th>
<th>Replication</th>
<th>Migration</th>
</tr>
</thead>
<tbody>
<tr>
<td>one.com</td>
<td>88.00%</td>
<td>97.00%</td>
<td>94.00%</td>
</tr>
<tr>
<td>two.com</td>
<td>92.00%</td>
<td>96.33%</td>
<td>94.00%</td>
</tr>
<tr>
<td>three.com</td>
<td>87.00%</td>
<td>96.33%</td>
<td>95.33%</td>
</tr>
<tr>
<td>four.com</td>
<td>98.00%</td>
<td>96.33%</td>
<td>93.66%</td>
</tr>
</tbody>
</table>

Throughput provided no conclusive results.
Analysis of Results

- Both *replication* and *migration* offer an improvement over taking *no action* upon detection of a resource stress situation.

- *Replication* offers a better improvement over *migration*.  

Overview

• Virtualization
• Uses of virtualization
• Virtualization in data centers
• VM replication

• VM memory management
VM Memory Management

VMs are allocated \textit{min} and \textit{max} amounts of memory.

We want more dynamism.

We are extending \textit{Golondrina} to allocate memory as needed.  
\textit{(Work in progress.)}
Virtualization is finding its way into many environments: industry, academia, government, HPC, Grid, Data Center...

Research topics are abound...
THANKS