An Empirical Study of the Impact of Cloud Patterns on Quality of Service (QoS) and Energy Consumption (Green IT)

Thibault Pere, Hugo Vautrin
Naouel Moha, Foutse Khomh (Tutors)
Motivations

Previous work

An Empirical Study of the Impact of Cloud Patterns on Quality of Service (QoS)

Shown:

Design Patterns can improve the QoS of a Cloud application

Are they also good in terms of Energy consumption?

References: An Empirical Study of the Impact of Cloud Patterns on Quality of Service (QoS)
Energy Consumption seen as an **Hardware Problem**

**Our Study**: Trying to solve it with Software solutions

**Focus On** **Design Patterns**
How many energy does a software consume?

<table>
<thead>
<tr>
<th></th>
<th>Master</th>
<th>Slaves 1,2 et 3</th>
<th>Slaves 4, 5, 6 et 7</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cpu</td>
<td>62.96</td>
<td>105.00</td>
<td>139.57</td>
<td>307.53</td>
</tr>
<tr>
<td>Mem</td>
<td>1.34</td>
<td>3.75</td>
<td>5.02</td>
<td>10.11</td>
</tr>
</tbody>
</table>

kilowatt hour (KWh)

3 LCD TV in eve mode 7d/7 - 24h/24 : 66 kWh
1 A+ class fridge 7d/7 - 24h/24 : 201 kWh
Null hypothesis $H_0$: Design Patterns doesn’t have any effect on Energy Consumption
Proxy

Different implementations... Random Round-Robin Custom
Message Queue

**Sharding**

Sharding logic:
- Route requests for tenant 1 to shard ...
- Route requests for tenant 55 to shard A
- Route requests for tenant 227 to shard C
- Route requests for tenant N to shard ...

References:
What do we need?

We need...

... tools to measure energy consumption!

... realistic test cases for a cloud environment

... a good architecture
Our Architecture

Hypervisor Type 1

TYPE 1
native
(bare metal)

TYPE 2
hosted
Our Architecture

Each VM:
- 2 Virtual CPU
- 1 Go RAM
- 20 Go DD

Replication

Replication

Sharding

Application

4 Virtual Machines

1 Virtual Machine

The VM:
- 4 CPU
- 8 Go RAM
- 50 Go DD

Each VM:
- 2 Virtual CPU
- 1 Go RAM
- 20 Go DD

Replication

Replication

Sharding

Application

4 Virtual Machines

1 Virtual Machine

Each VM:
- 2 Virtual CPU
- 1 Go RAM
- 20 Go DD

Replication

Replication

Sharding

Application

4 Virtual Machines

1 Virtual Machine
Need basic information about hardware components to calculate
PowerAPI - How it works
Is PowerAPI a good tool?

**First Test : Does the measure of PowerAPI affect tests?**

**Aim :**
Be sure PowerAPI doesn’t distort our tests.

**Environement :**
- 1 VM

**Objects monitored :**
- MySQL
- Eclipse (PowerAPI)

---

Start Of Monitoring

10 000 Inserts
Wait 1 Minute
1 Select ALL
10 000 Inserts
Wait 1 Minute
1 Select ALL
Wait 3 Minutes

Test Case

End Of Monitoring
Is PowerAPI a good tool?

Energy consumption:
PowerAPI is negligible compared to MySQL
Is PowerAPI a good tool?

PowerAPI doesn’t interfere with our measures

Memory Consumption
Our Study: Energy Consumption

RQ1: Has the Proxy Pattern an effect on Energy Consumption?

RQ2: Has the Sharding Pattern an effect on Energy Consumption?

RQ3: Has the Sharding + MQ Pattern an effect on Energy Consumption?
A realistic test case (Proxy) - RQ1

Realistic Scenario for our tests (Cloud Application and Environment)

- **Number of Repetition**: 3
- **Number of Clients**: {500; 1500; 3000}
- **Environment**: 1 MASTER - 4 SLAVES (2:2)
- **Databases**: 10 000 movies

---

**Start Of Monitoring**

First set of Clients
Wait 30 Seconds

Second set of Clients
Wait 30 Seconds

Third set of Clients
Wait 30 Seconds

**End Of Monitoring**

**Test Case**

**Connexion** → Reading the Menu → **Consulting 10 Movies** → Thinking, chatting…
→ **Consulting 5 over Movies** → Taking a Decision → **Renting a Movie**
→ Waiting for validation → **Logging off**
No Proxy is better than any implementation of Proxy
Proxy’s results - RQ1

500 Clients

\[ \Delta \text{(Custom - No Proxy)} = \text{a classical use of an hair dryer} \]

<table>
<thead>
<tr>
<th>CPU</th>
<th>Random</th>
<th>Round Robin</th>
<th>Custom</th>
<th>No Proxy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control</td>
<td>60,84</td>
<td>60,31</td>
<td>61,95</td>
<td>52,71</td>
</tr>
<tr>
<td>Whole Cloud</td>
<td>23,57</td>
<td>24,30</td>
<td>23,46</td>
<td>22,00</td>
</tr>
<tr>
<td>Total</td>
<td>84,41</td>
<td>84,62</td>
<td>85,41</td>
<td>74,70</td>
</tr>
</tbody>
</table>

1500 Clients

\[ \Delta \text{(Custom - No Proxy)} = \text{a classical use of a Coffee Maker} \]

<table>
<thead>
<tr>
<th>CPU</th>
<th>Random</th>
<th>Round Robin</th>
<th>Custom</th>
<th>No Proxy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control</td>
<td>83,53</td>
<td>81,52</td>
<td>87,95</td>
<td>53,89</td>
</tr>
<tr>
<td>Whole Cloud</td>
<td>79,89</td>
<td>79,43</td>
<td>81,77</td>
<td>74,67</td>
</tr>
<tr>
<td>Total</td>
<td>163,41</td>
<td>160,96</td>
<td>169,73</td>
<td>128,56</td>
</tr>
</tbody>
</table>

3000 Clients

\[ \Delta \text{(Custom - No Proxy)} = \text{LCD TV en eve mode during a year} \]

<table>
<thead>
<tr>
<th>CPU</th>
<th>Random</th>
<th>Round Robin</th>
<th>Custom</th>
<th>No Proxy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control</td>
<td>87,15</td>
<td>90,7</td>
<td>92,78</td>
<td>70,55</td>
</tr>
<tr>
<td>Whole Cloud</td>
<td>152,37</td>
<td>149,7</td>
<td>154,2</td>
<td>154,46</td>
</tr>
<tr>
<td>Total</td>
<td>239,52</td>
<td>240,4</td>
<td>246,98</td>
<td>225,01</td>
</tr>
</tbody>
</table>
A realistic test case (Sharding) - RQ2

Realistic Scenario for our tests (Cloud Application and Environment)

*Number of Repetition* : 3; *Number of Clients* : {500; 1500; 3000}
*Environment* : 4 SHARDS
*Databases* : 10 000 movies

Cloud Application
Rent a movie!

Connexion → Reading the Menu → Consulting 10 Movies → Thinking, chatting…
→ Consulting 5 over Movies → Taking a Decision → Renting a Movie
→ Waiting for validation → Logging off

Start Of Monitoring
First set of Clients
Wait 30 Seconds
Second set of Clients
Wait 30 Seconds
Third set of Clients
Wait 30 Seconds
End Of Monitoring

Test Case
Modulo is better than any implementation of Sharding
Sharding’s results - RQ2

500 Clients

<table>
<thead>
<tr>
<th>CPU</th>
<th>Modulo</th>
<th>Look up</th>
<th>Consistent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control</td>
<td>91.67</td>
<td>60.62</td>
<td>76.24</td>
</tr>
<tr>
<td>Whole Cloud</td>
<td>15.80</td>
<td>15.42</td>
<td>15.76</td>
</tr>
<tr>
<td>Total</td>
<td>107.47</td>
<td>76.05</td>
<td>92.00</td>
</tr>
</tbody>
</table>

\[ \Delta(\text{Look Up - Modulo}) = \text{Lamp energy saving bulb} \]

\[ \Delta(\text{Modulo - Look Up}) = \text{LCD TV in eve mode during a year} \]

1500 Clients

<table>
<thead>
<tr>
<th>CPU</th>
<th>Modulo</th>
<th>Look up</th>
<th>Consistent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control</td>
<td>79.03</td>
<td>98.74</td>
<td>91.56</td>
</tr>
<tr>
<td>Whole Cloud</td>
<td>62.60</td>
<td>63.42</td>
<td>62.97</td>
</tr>
<tr>
<td>Total</td>
<td>141.63</td>
<td>162.16</td>
<td>154.52</td>
</tr>
</tbody>
</table>

3000 Clients

<table>
<thead>
<tr>
<th>CPU</th>
<th>Modulo</th>
<th>Look up</th>
<th>Consistent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control</td>
<td>95.21</td>
<td>107.97</td>
<td>106.60</td>
</tr>
<tr>
<td>Whole Cloud</td>
<td>131</td>
<td>129.55</td>
<td>124.94</td>
</tr>
<tr>
<td>Total</td>
<td>226.08</td>
<td>237.51</td>
<td>231.55</td>
</tr>
</tbody>
</table>

\[ \Delta(\text{Modulo - Look Up}) = \text{a classical use of an hair dryer} \]
Sharding’s results with MQ - RQ3

3000 Clients:

\[ \Delta (\text{Look Up + MQ} - \text{Look Up}) = \text{a classical use of an electric cooker} \]

Same results for Modulo and Consistent

References: An Empirical Study of the Impact of Cloud Patterns on Quality of Service (QoS)
Summary

Null hypothesis $H_0$: Design Patterns doesn’t have any effect on Energy Consumption

$H_0$ is false

- Design Patterns have an effet on Energy Consumption
  - Trade-off (QoS vs Green IT)
Futur Work ...

Combinations
- Message Queue + Proxy
- Proxy + Sharding
- Proxy + Sharding + Message Queue

Giving tools for engineers and developers to choose the best approach for their cloud application function of their needs (latency, energy, etc)
Null hypothesis H0: Design Patterns doesn’t have any effect on Energy Consumption

Shown:
Design Patterns can improve the QoS of a Cloud application

Are they also good in terms of Energy consumption?

Design Patterns of our study
(databases oriented)

Proxy
Message Queue
Sharding
...and their combinations

PowerAPI

Comparison of Proxy Algorithms

Δ(Custom - No Proxy) = LCD TV en eee mode during a year