

# 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)**



POLYTECHNIQUE  
MONTRÉAL

LE GÉNIE  
EN PREMIÈRE CLASSE



POLYTECH<sup>®</sup>  
MONTPELLIER

# Motivations

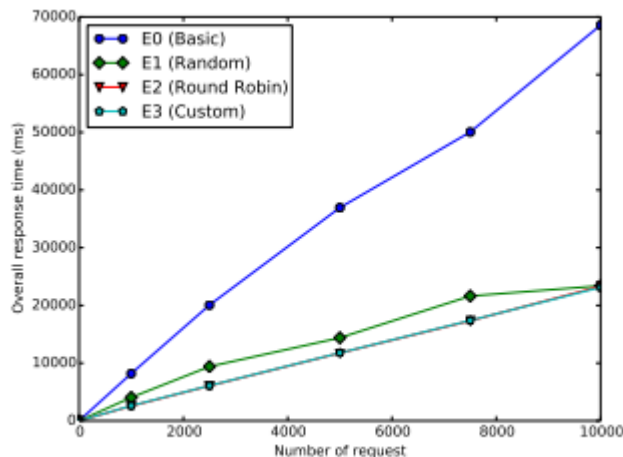
---

## 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 ?**

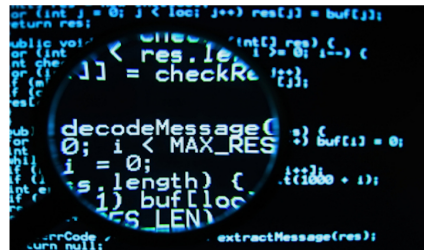
# Context

---



Energy Consumption seen as an **Hardware Problem**

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



Focus On **Design Patterns**

# How many energy a software consume ?

	Master	Slaves 1,2 et 3	Slaves 4, 5, 6 et7	Total
Cpu	62,96	105,00	139,57	307,53
Mem	1,34	3,75	5,02	10,11

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**

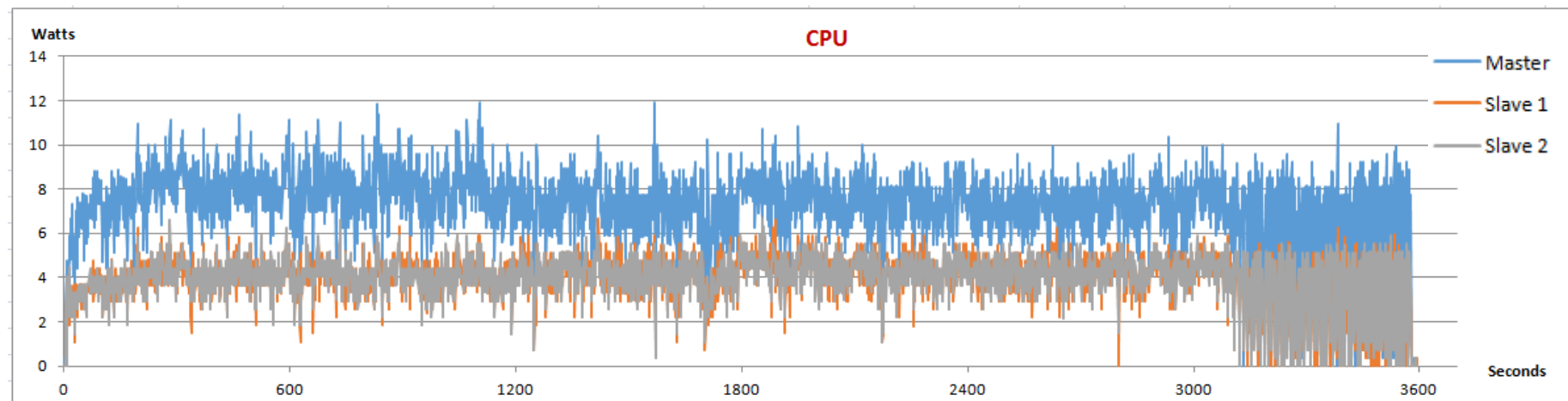
Start Of Monitoring

Wait 1 Minute

Start 1h Insert script

Wait 1 Minute

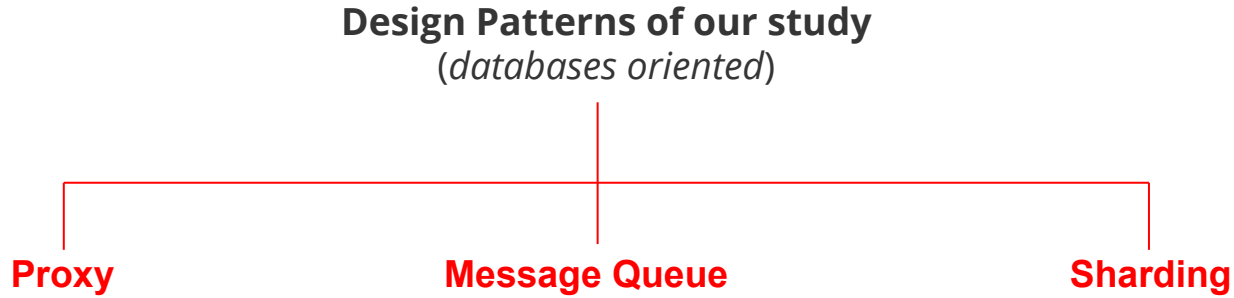
End Of Monitoring



# Hypothesis

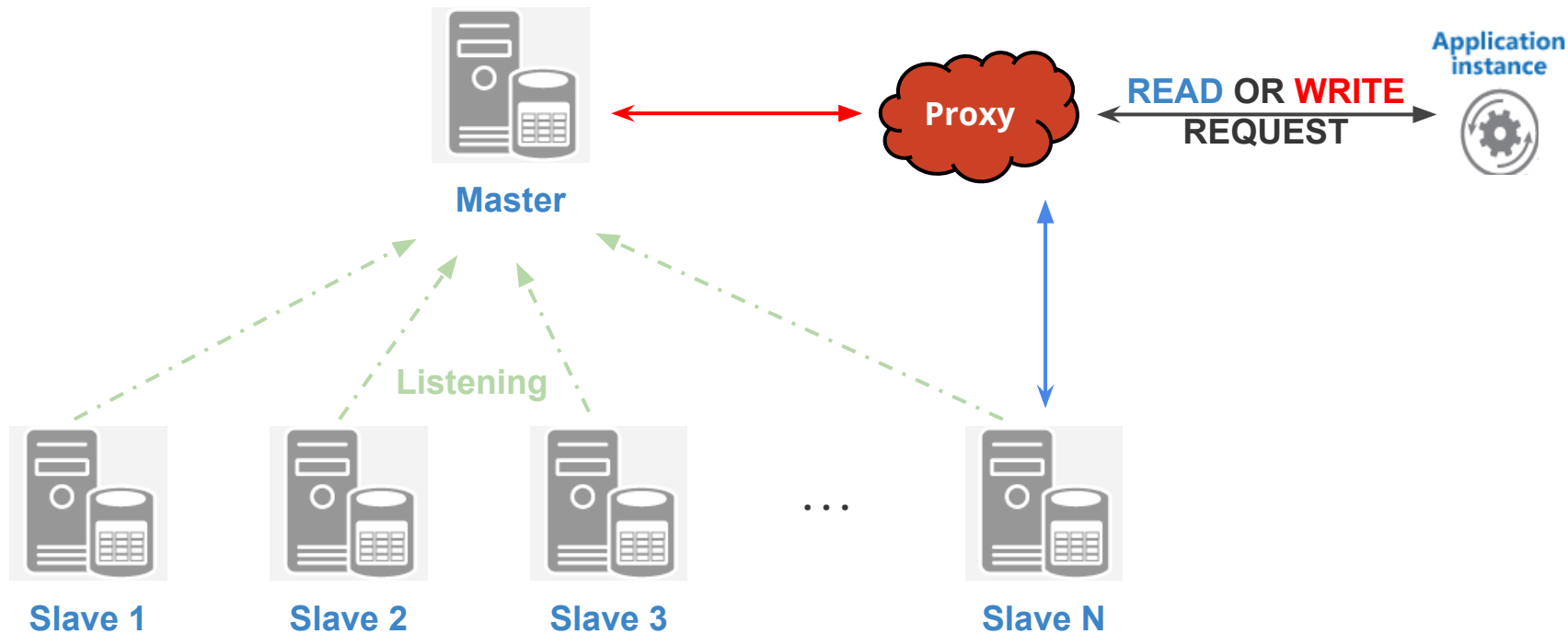
---

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



**...and their combinations**

# Proxy



Different implementations...

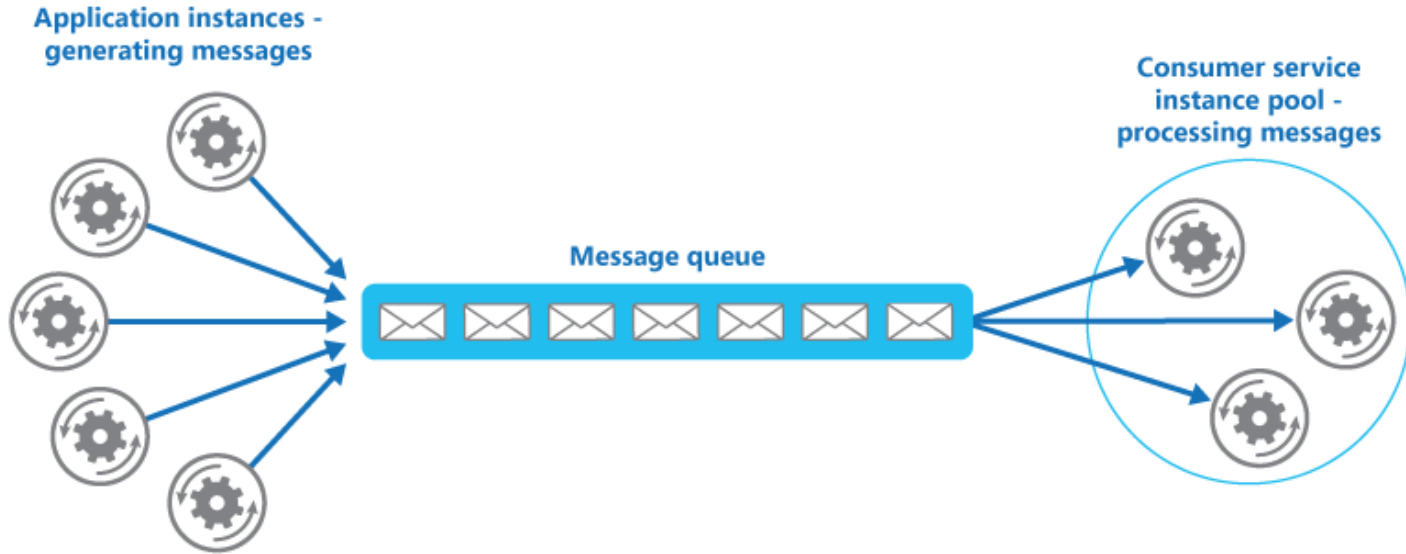
Random

Round-Robin

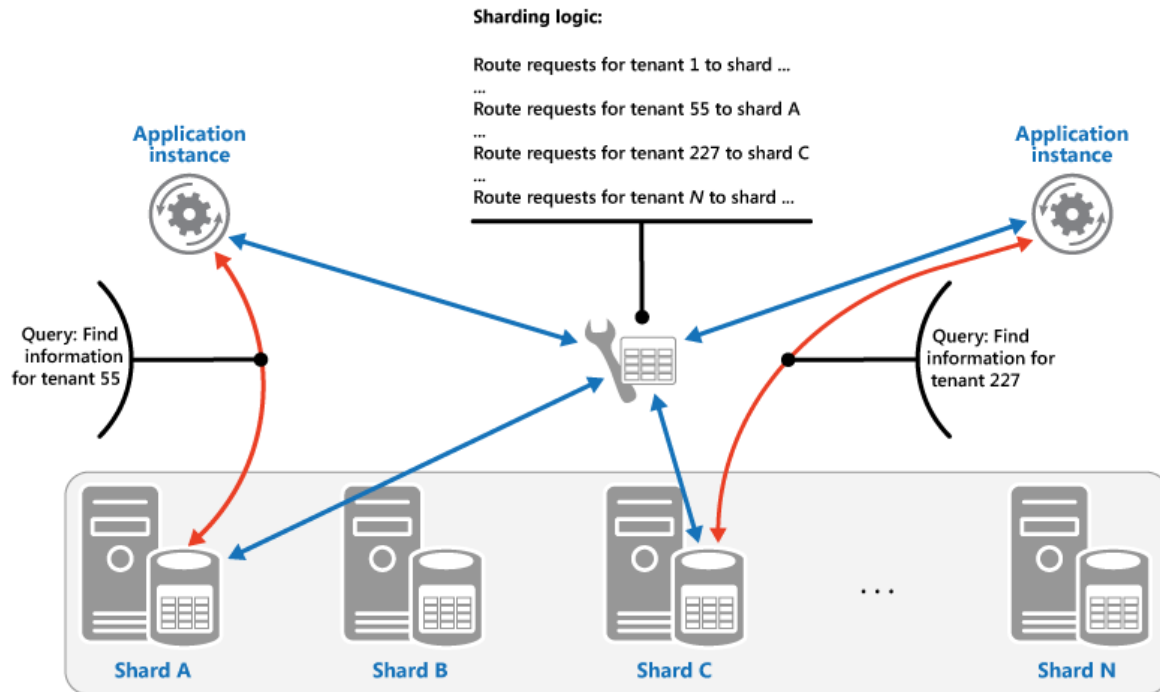
Custom

# Message Queue

---



# Sharding



Different implementations...

Modulo

Look Up

Consistent Hashing



# What do we need ?

---



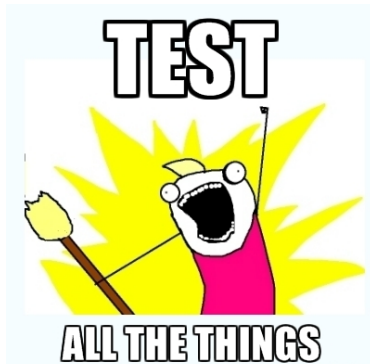
**We need...**

**... a good architecture**



**... tools to measure energy consumption !**

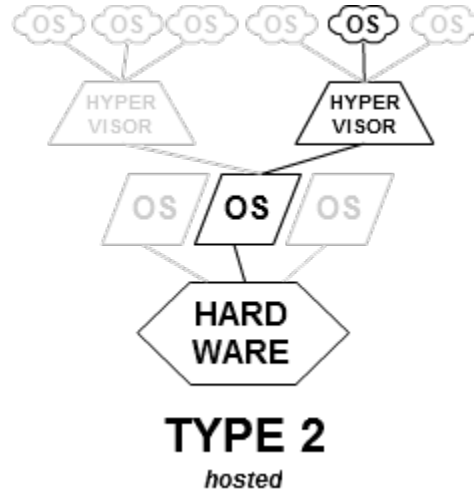
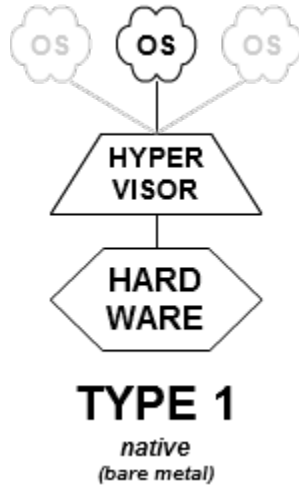
**... realistic test cases for a cloud environment**



# Our Architecture

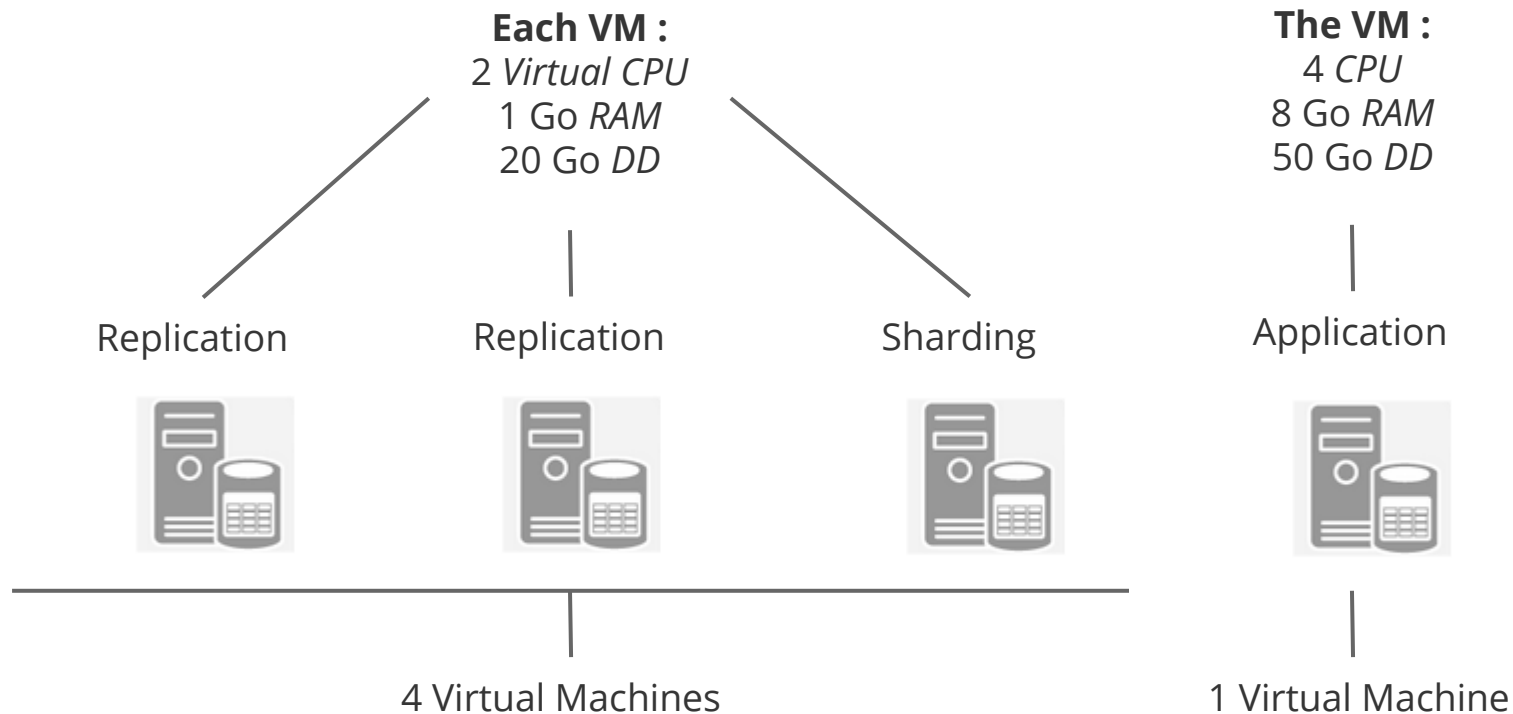
---

## Hypervisor Type 1



# Our Architecture

---



# PowerAPI

---



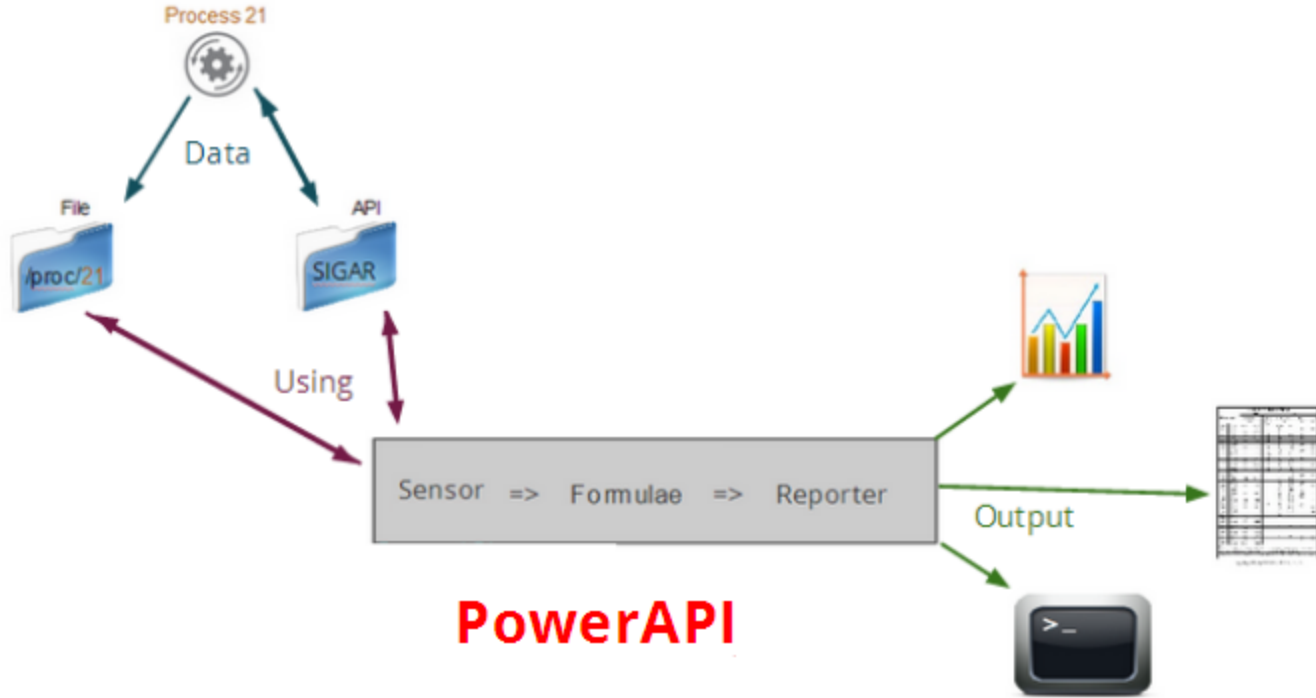
**maven**

Need basic information about  
hardware components to  
calculate



*inria*

# 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.

**Environnement :**

- 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

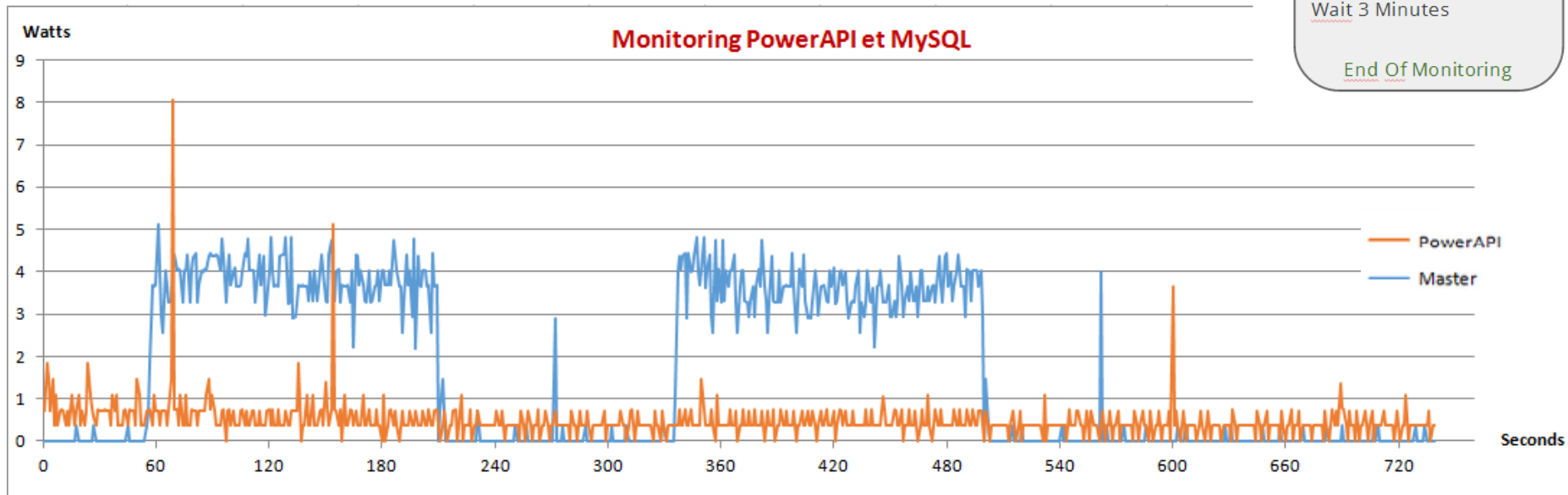
Wait 3 Minutes

End Of Monitoring

**Test Case**

# Is PowerAPI a good tool?

Energy consumption :  
PowerAPI is negligible compared to MySQL



Start Of Monitoring

10 000 Inserts

Wait 1 Minute

1 Select ALL

10 000 Inserts

Wait 1 Minute

1 Select ALL

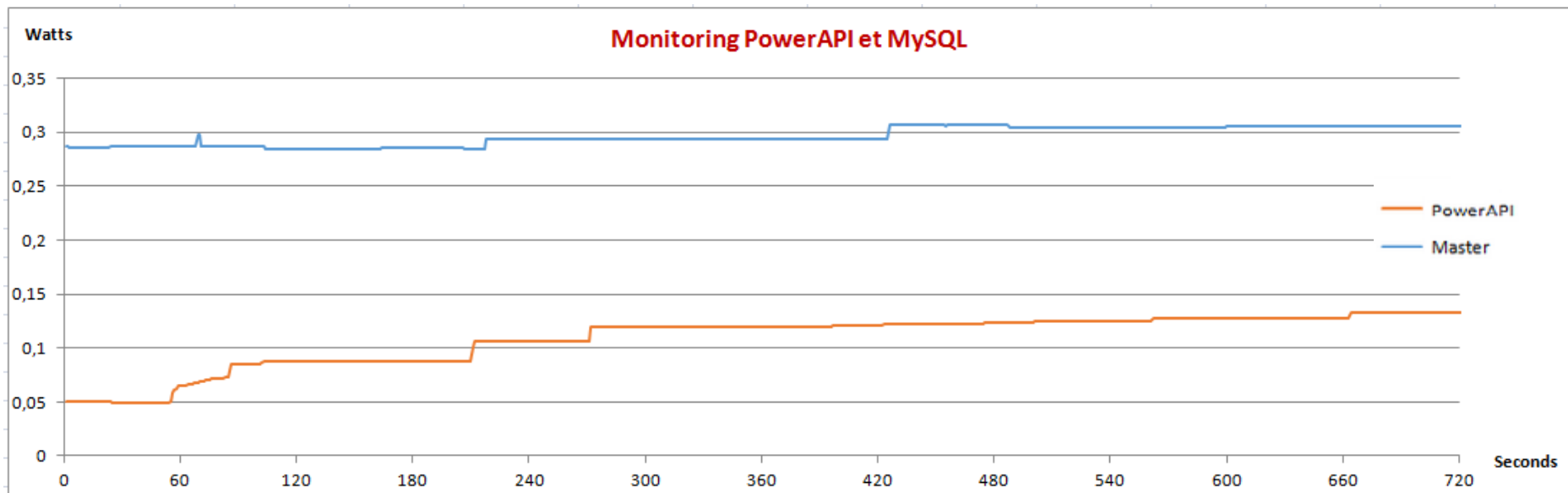
Wait 3 Minutes

End Of Monitoring

CPU Consumption

# Is PowerAPI a good tool?

PowerAPI doesn't interfere with our measures



Memory Consumption



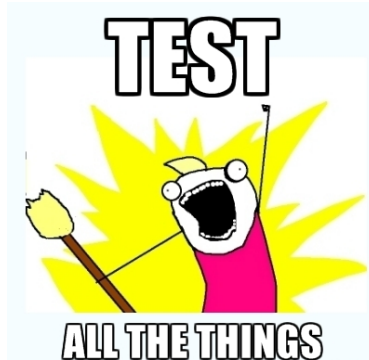
# Our Study : Energy Consumption

---

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

**RQ2:** Has the ShardingPattern an effect on Energy Consumption ?

**RQ3:** Has the Sharding + MQ Pattern an effect on Energy Consumption ?



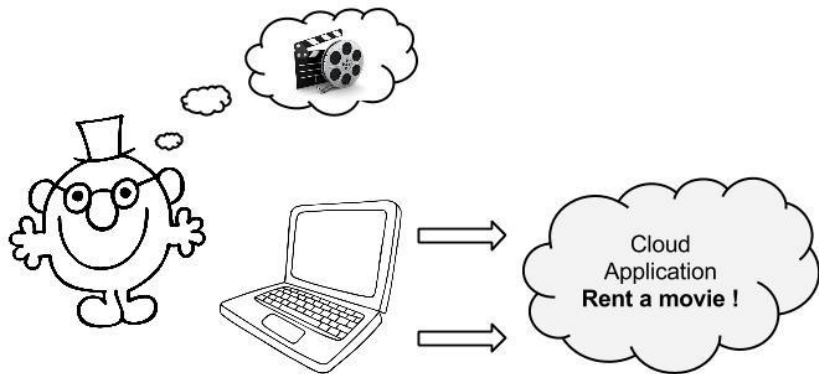
# A realistic test case (Proxy) - RQ1

## Realistic Scenario for our tests (Cloud Application and Environment)

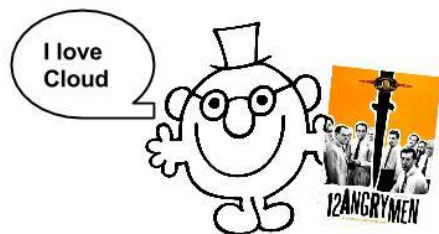
*Number of Repetition : 3; Number of Clients : {500; 1500; 3000}*

*Environment : 1 MASTER - 4 SLAVES (2:2)*

*Databases : 10 000 movies*



**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

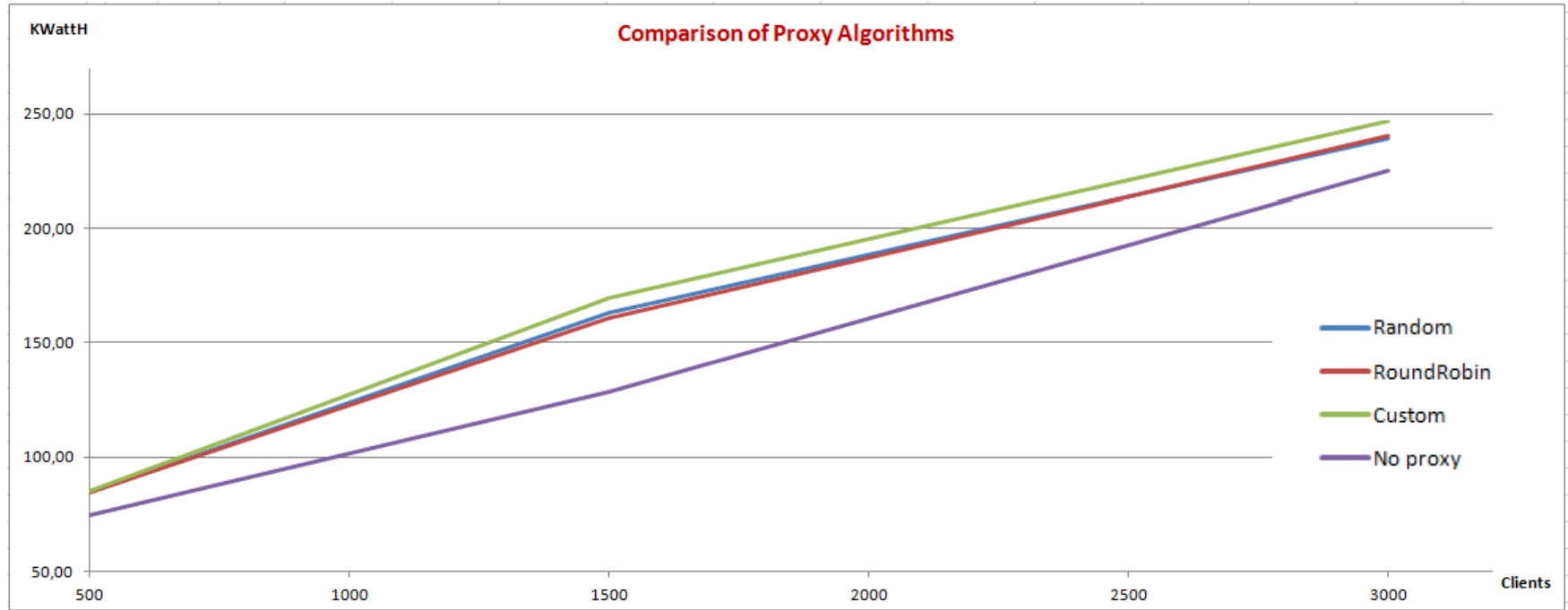
Wait 30 Seconds

End Of Monitoring

**Test Case**

# Proxy's results - RQ1

---



**No Proxy** is better than any implementation of Proxy

# Proxy's results - RQ1

500 Clients

CPU	Random	Round Robin	Custom	No Proxy
Control	60,84	60,31	61,95	52,71
Whole Cloud	23,57	24,30	23,46	22,00
Total	84,41	84,62	85,41	74,70

$\Delta$ (Custom - No Proxy)

=

a classical use of an hair  
dryer

$\Delta$ (Custom - No Proxy)

=

a classical use of a Coffee  
Maker

1500 Clients

CPU	Random	Round Robin	Custom	No Proxy
Control	83,53	81,52	87,95	53,89
Whole Cloud	79,89	79,43	81,77	74,67
Total	163,41	160,96	169,73	128,56

3000 Clients

CPU	Random	Round Robin	Custom	No Proxy
Control	87,15	90,7	92,78	70,55
Whole Cloud	152,37	149,7	154,2	154,46
Total	239,52	240,4	246,98	225,01

$\Delta$ (Custom - No Proxy)

=

LCD TV en eve mode  
during a year

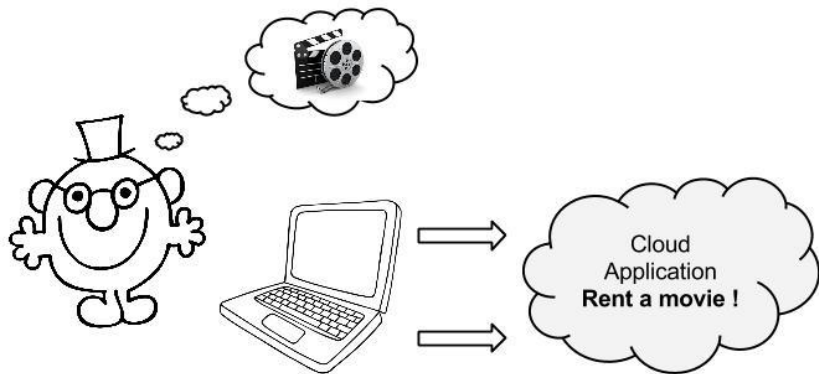
# A realistic test case (Sharding) - RQ2

## Realistic Scenario for our tests (Cloud Application and Environment)

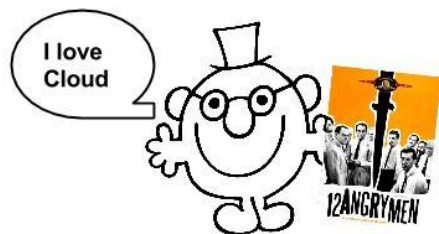
*Number of Repetition : 3; Number of Clients : {500; 1500; 3000}*

*Environment : 4 SHARDS*

*Databases : 10 000 movies*



**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

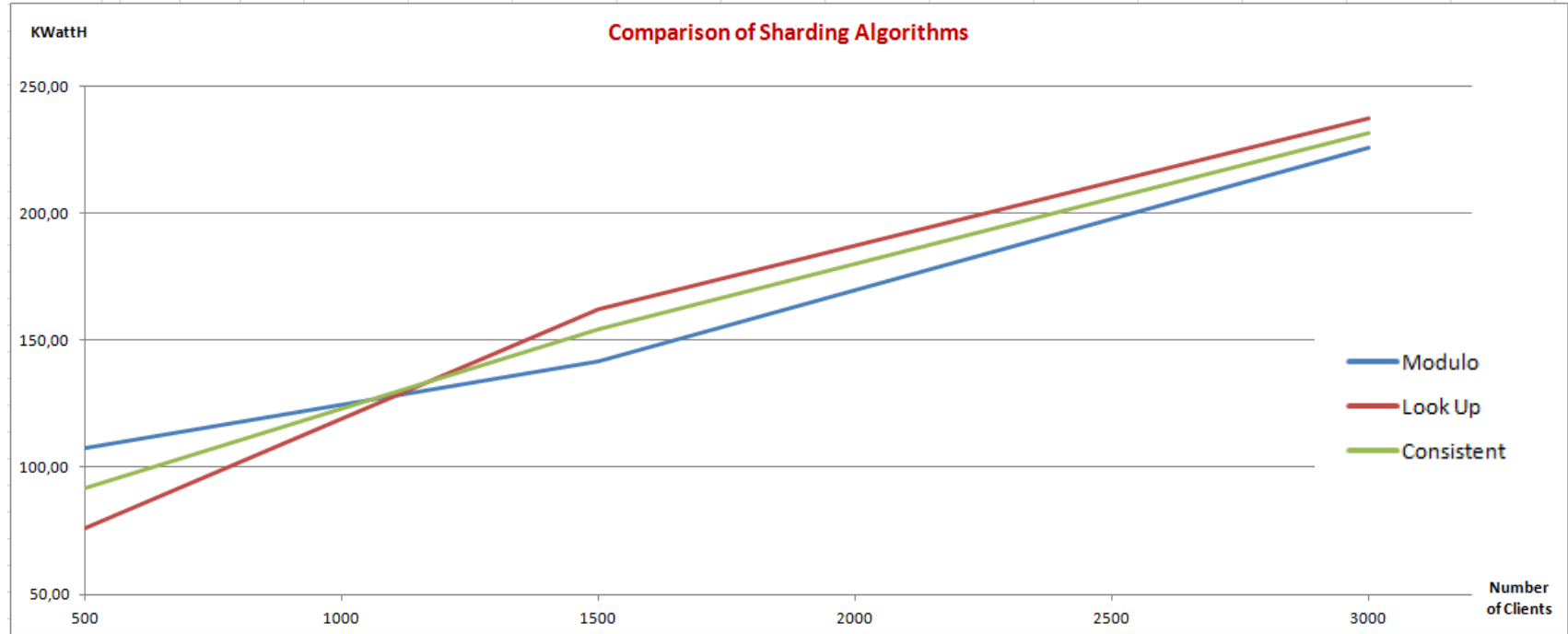
Third set of Clients

Wait 30 Seconds

End Of Monitoring

**Test Case**

# Sharding's Result - RQ2



**Modulo** is better than any implementation of Sharding

# Sharding's results - RQ2

500 Clients

CPU	Modulo	Look up	Consitent
Control	91,67	60,62	76,24
Whole Cloud	15,80	15,42	15,76
Total	107,47	76,05	92,00

$\Delta(\text{Look Up} - \text{Modulo})$

=

Lamp energy saving bulb

$\Delta(\text{Modulo} - \text{Look Up})$

=

LCD TV in eve mode  
during a year

1500 Clients

CPU	Modulo	Look up	Consitent
Control	79,03	98,74	91,56
Whole Cloud	62,60	63,42	62,97
Total	141,63	162,16	154,52

3000 Clients

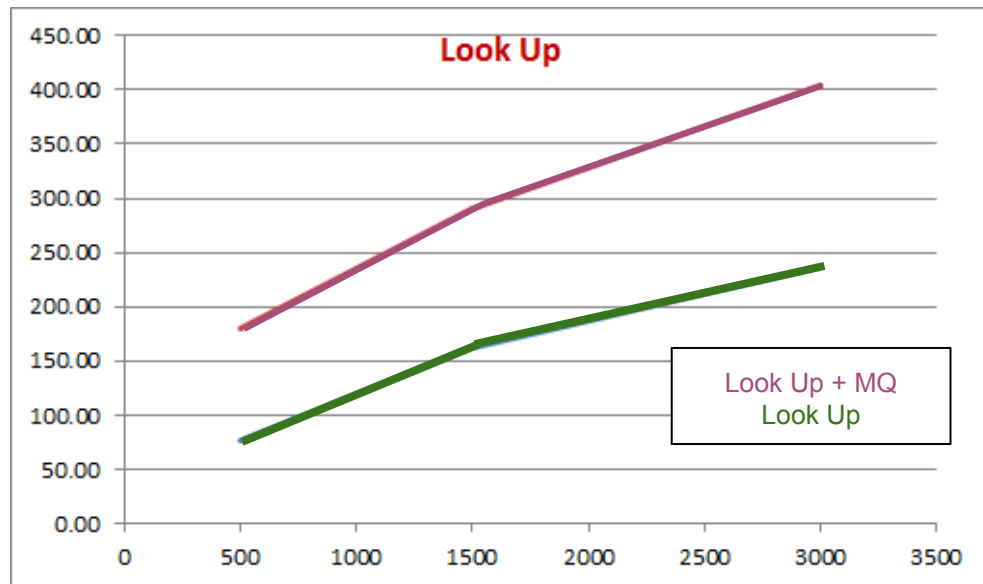
CPU	Modulo	Look up	Consitent
Control	95,21	107,97	106,60
Whole Cloud	131	129,55	124,94
Total	226,08	237,51	231,55

$\Delta(\text{Modulo} - \text{Look Up})$

=

a classical use of an hair  
dryer

# Sharding's results with MQ - RQ3



Same results for Modulo and Consistent

3000 Clients :

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

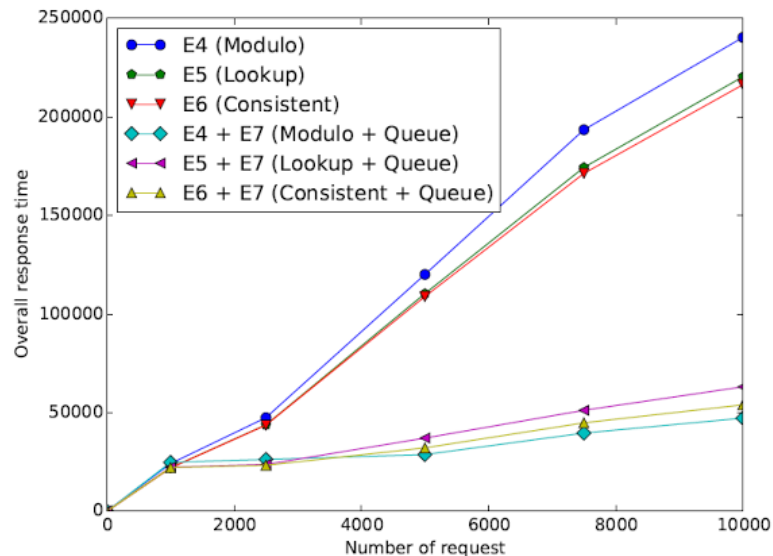


Fig. 5. Insert film with Local Sharding-Based Router and Priority Queue



# Summary

---

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

***$H_0$  is false***

- Design Patterns have an effect on Energy Consumption
  - Trade-off (QoS vs Green IT)

# Futur Work ...

---

## Combinations

- *Message Queue + Proxy*
- *Proxy + Sharding*
- *Proxy + Sharding + Message Queue*



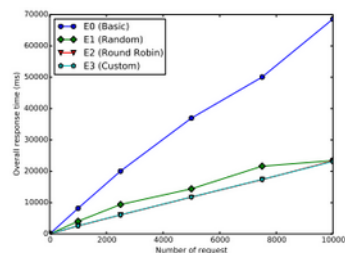
**Giving tools for engineers and developers to choose the best approach for their cloud application function of their needs (latency, energy, etc)**

# Previous work

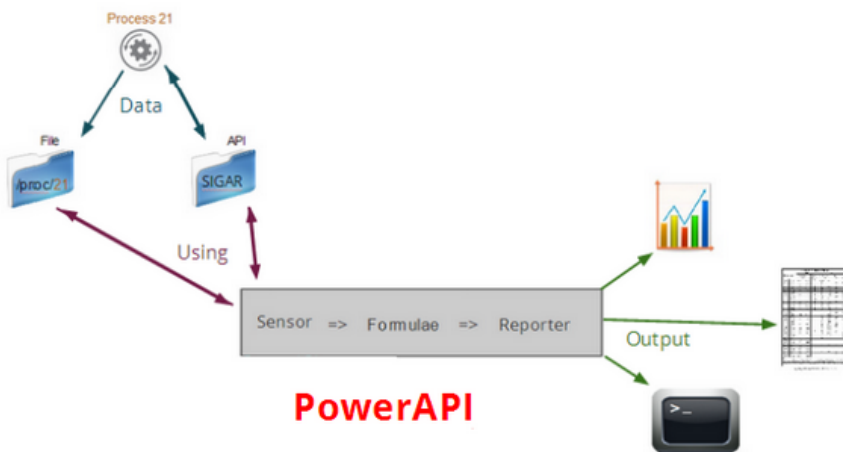
An Empirical Study of the Impact of Cloud Patterns on Quality of Service (QoS)

Shown :

Design Patterns can improve the QoS of a Cloud application



Are they also good in terms of Energy consumption ?



**Null hypothesis H0 :** Design Patterns doesn't have any effect on Energy Consumption

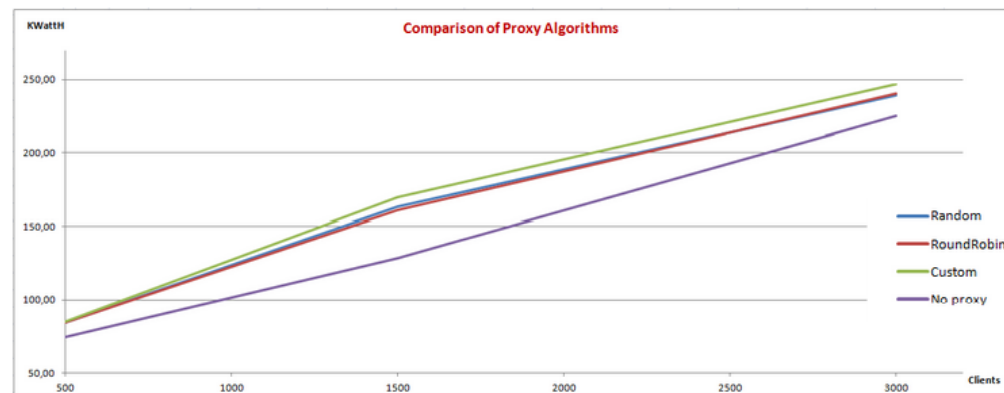
Design Patterns of our study  
(databases oriented)

Proxy

Message Queue

Sharding

...and their combinations



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

3000 Clients