Performance characterization of black boxes with self-controlled load injection for simulation-based sizing

Ahmed Harbaoui[1,2] Bruno Dillenseger[1] Jean-Marc Vincent[2]

{Ahmed.Harbaoui, Bruno.Dillenseger} @orange-ftgroup.com, Jean-Marc.Vincent@imag.fr

[1] France Telecom R&D

[2] Laboratoire LIG, Projet MESCAL

Presented in CFSE 2008

Introduction

- Sizing and capacity planning are key issues that must be addressed by anyone wanting to ensure a distributed system will sustain an expected workload.
- Example : Deployment of new service for a million of users.
- Questions: How much server to use? Middleware ? … to support a given load.

- In many cases, sizing is done without a defined methodology
- Problem : Empirical sizing:
  - Loss of a huge amount of money

    OR
  - Performance problems

Problem Position: Sizing Complex/Distributed systems

- Full-scale test

Systems are more and more :

1. Distributed 2. Complex 3. A lack of information

→ This solution may be too costly and almost impossible

- Simulation

→ Must have a **Model** of system

Black boxes

Performance Modeling of system as Black boxes

Outline

1. Sizing Problem
2. Modeling for performance analysis
   1. How to model black boxes
   2. Our methodology to model black boxes
3. Saturation and stability
4. Experimental framework
   1. CLIF
   2. Infrastructure
   3. Experience
5. Future work

How to model Black boxes

- How to model a system :
  - Analytical approach Mathematical model (automata, queuing network, …)
  - Simulation A simplified model for the system by using suitable software
  - Workload Emulation direct measurements and analysis are carried out on the system.

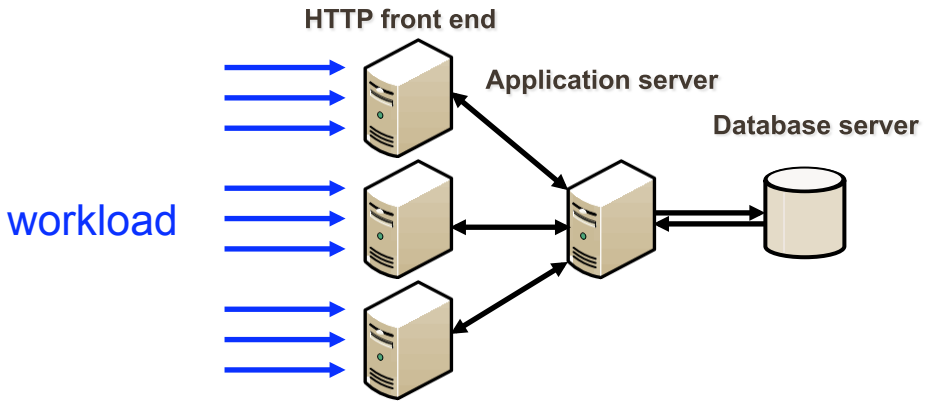
Black boxes

Workload emulation + prototype black box

Our methodology to model black boxes

- Our methodology consists in several steps to produce Performance Model:

1. Choosing performance indexes
2. Black boxes decomposition
3. Workload specification
4. Instrumentation
5. Modeling



Methodology (1/5): Choosing performance indexes

- The indexes are the different characteristics that impact system performance.

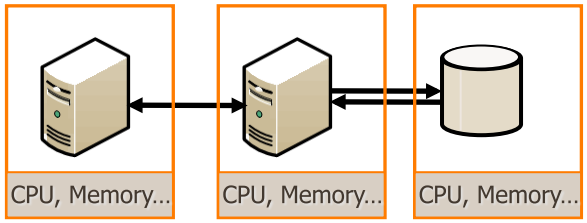
Example : For 3 tiers applications indexes are:

1. End-to-end response time,
2. Throughput in requests per time unit,
3. Number of customers per time unit,

Users aspects

4. CPU load, memory, network usage.

Resources aspects



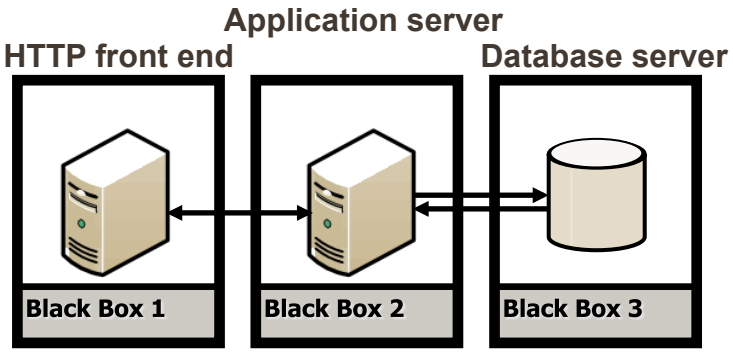
Problem : A great number of indexes ➔ **a huge amount of time**.

Approach : Factor analysis

Methodology (2/5): **System decomposition into black boxes**

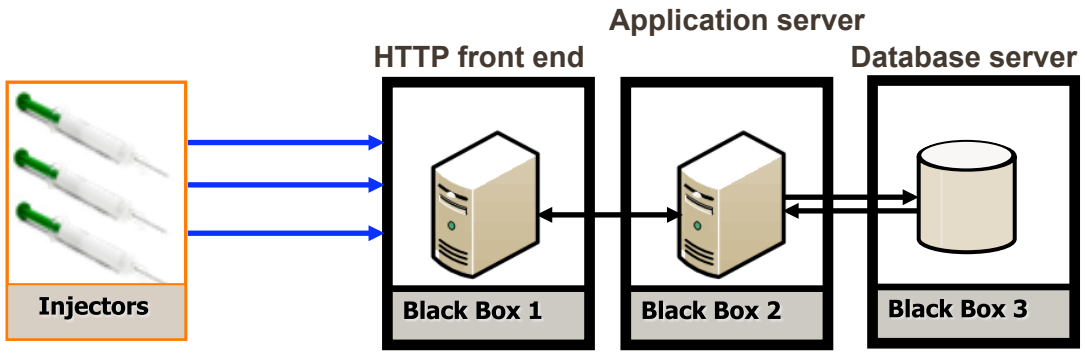
- Try to divide the system into black boxes network according to:

1. Application Architecture : multi-tiers server, component based application.
2. Locality : distributed application.
3. …



Methodology (3/5):Workload specification

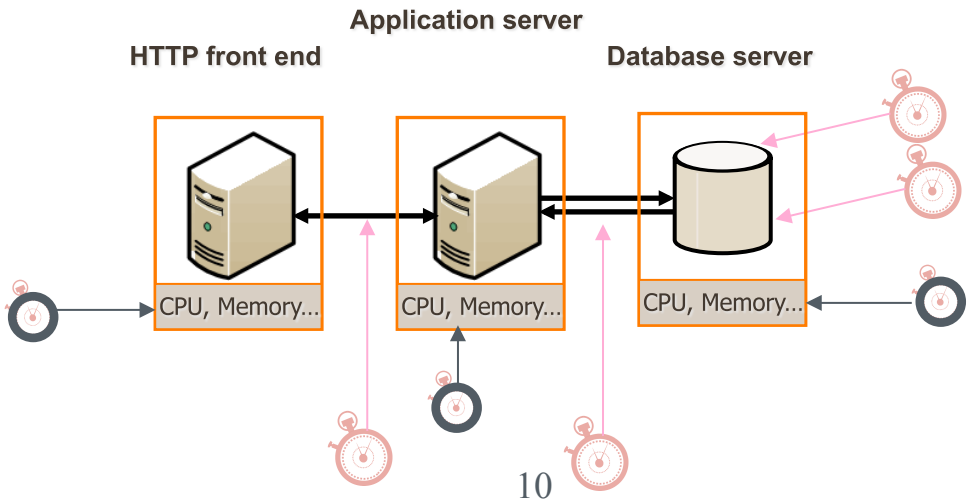
- According to black boxes decomposition and performance indexes choosing We define the workload to apply.
  - **Synthetic workload.**
  - **Replay a real workload.**
- For good qualification → the test load must be as close as possible to the real load.



Methodology (4/5): Instrumentation

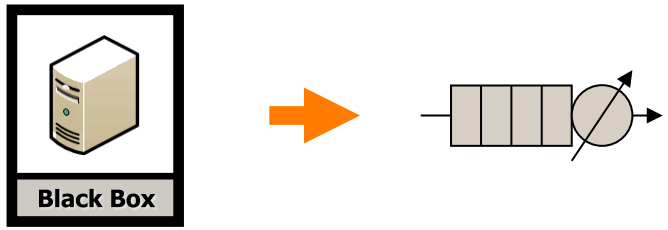
- Instrumentation is define according to black boxes decomposition and performance indexes choosing.

Instrumentation deals with monitoring and measuring the use of resources (CPU, memory, …) by placing probes in different parts of the system under test.



Methodology (5/5): Modeling

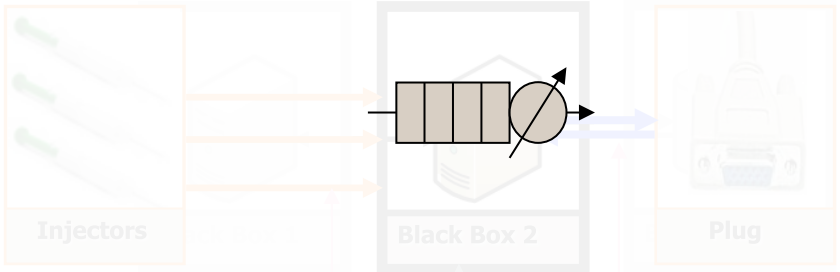
- Queuing Network
  1. Adapted for performance study.
  2. Adapted to resource consumption : Server and Queue.
  3. Adapted to Black boxes decomposition.
- Our goal is to model each black box by a queue.

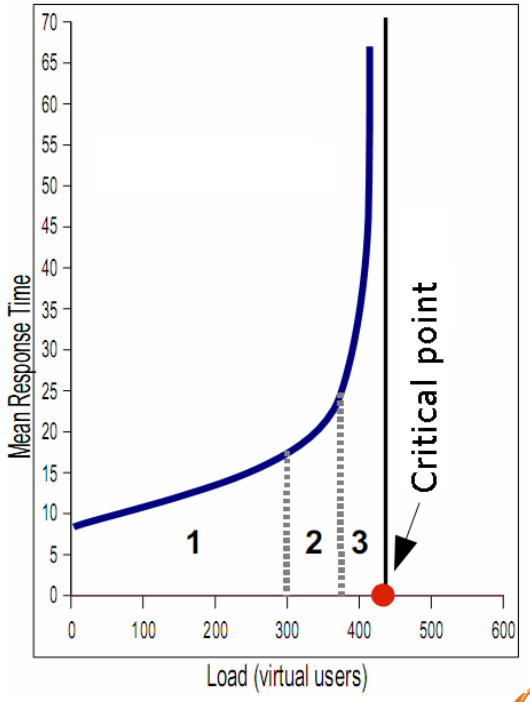


- 3 type of black boxes (load-dependent black boxes, load-independent black boxes, pure delay black boxes).

Modeling : Isolation & characterization of black box

- In the case of several interacting black boxes:
- **Software-plugs** : They replace interactions of the tested black box with other black boxes while conserving a constant value for performance parameters of interest.





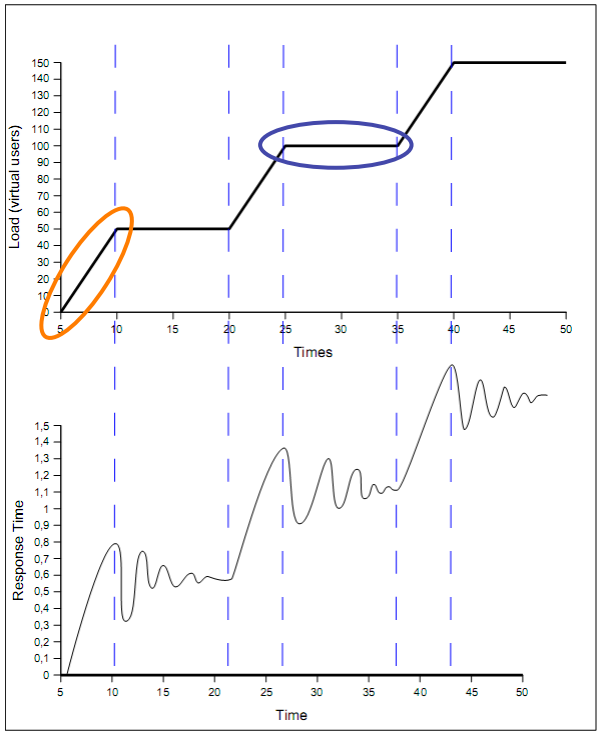
Saturation and stability

- To extract the maximum performance, all measurements should be done just before saturation.

- To reach saturation point, we must ensure that the system is stable each step.

- Empirical and Manual process

- This method is could take a **huge time**

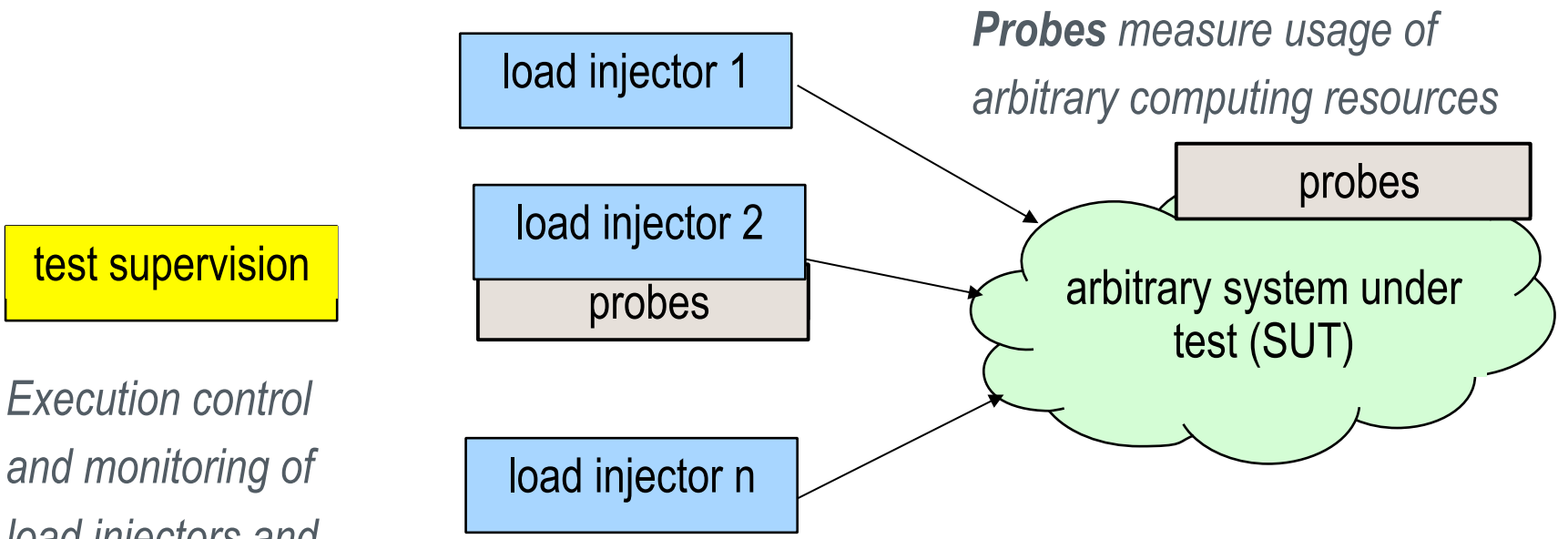


→ Autonomous critical point research

Outline

1. Sizing Problem
2. Modeling for performance analysis
   1. How to model black boxes
   2. Our methodology to model black boxes
3. Saturation and stability
4. Experimental framework
   1. CLIF
   2. Infrastructure
   3. Experience
5. Future work

CLIF Load Injection Framework



***Probes** measure usage of arbitrary computing resources*

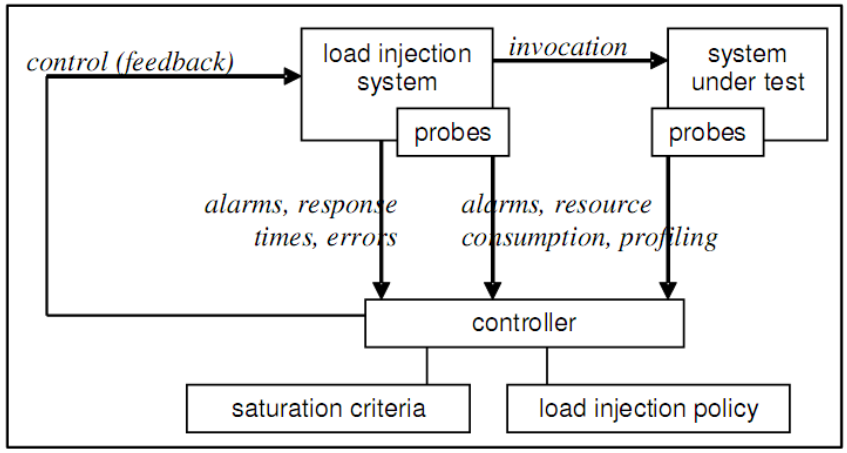
Execution control and monitoring of load injectors and resource probes.

***Load injectors** :*

- *send requests, wait for replies, measure response times*
- *according to a given scenario*
- *for example, emulating the load of a number of real users (through so-called virtual users)*

Component based infrastructure for autonomic Computing

- To experiment our methodology, we propose a practical component based infrastructure that fits:
  1. Genericity
  2. Autonomy
- Autonomic computing: is the principle of using computing power to get computing systems autonomously manage their complexity.



A self-regulated load injection experiment (1)

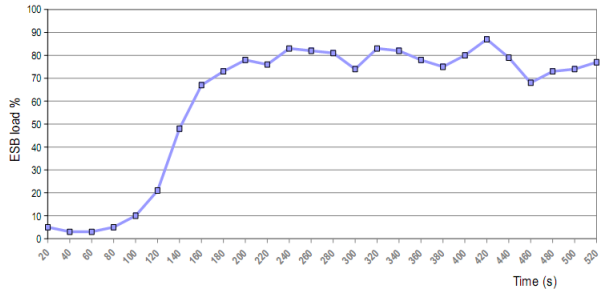
- Goal of experiment: find the ESB saturation limit with looped load injection system according to saturation criterion.
- SUT: **E**nterprise **S**ervice **B**us, a request broker used in Service Oriented Architectures to support mediation features such as accounting, routing, logging, security, management of service level agreement, etc.
- Black box: ESB.
- Workload: SOAP requests.
- Services: Our software plugs, i.e. dummy services that reply to requests with a constant response time, whatever the incoming workload.

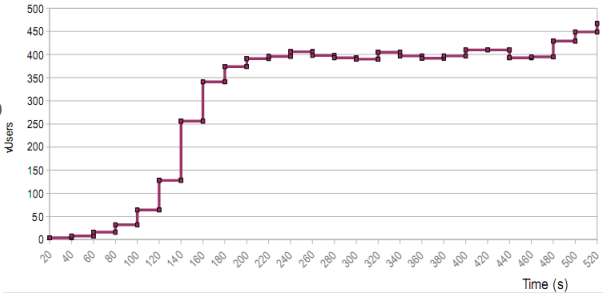
A self-regulated load injection experiment (2)

- 4 load injectors
- A controller distributed on 5 distinct computers

(Intel bi-Xeon or AMD bi-Opteron, 2 or 3 GB RAM, Gb/s Ethernet, Linux kernel 2.6.15-1-686-smp).

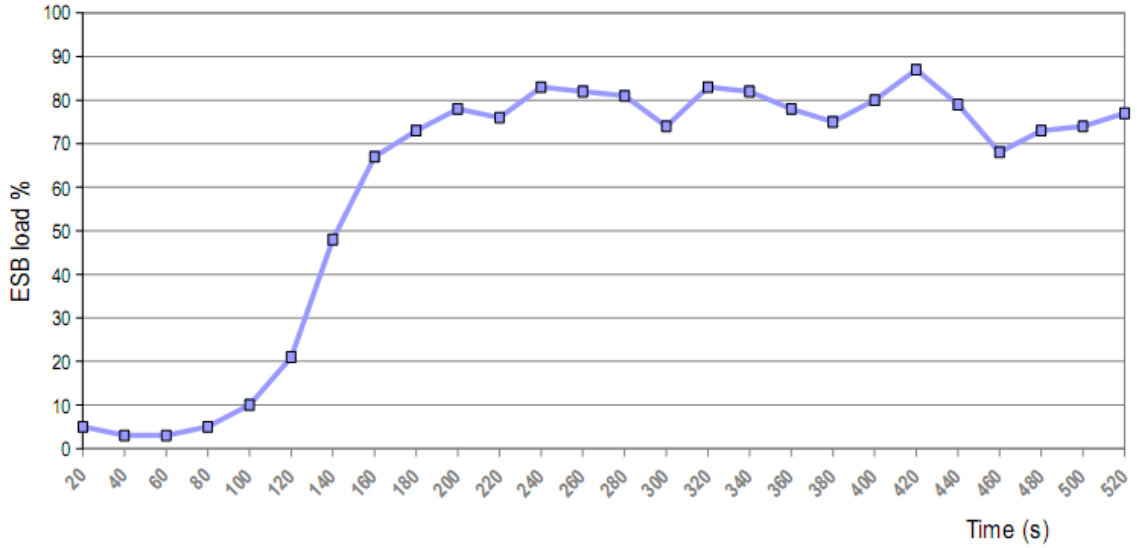
- The ESB load probe is hosted on a 6th computer and gets information from the ESB platform's SNMP agent.

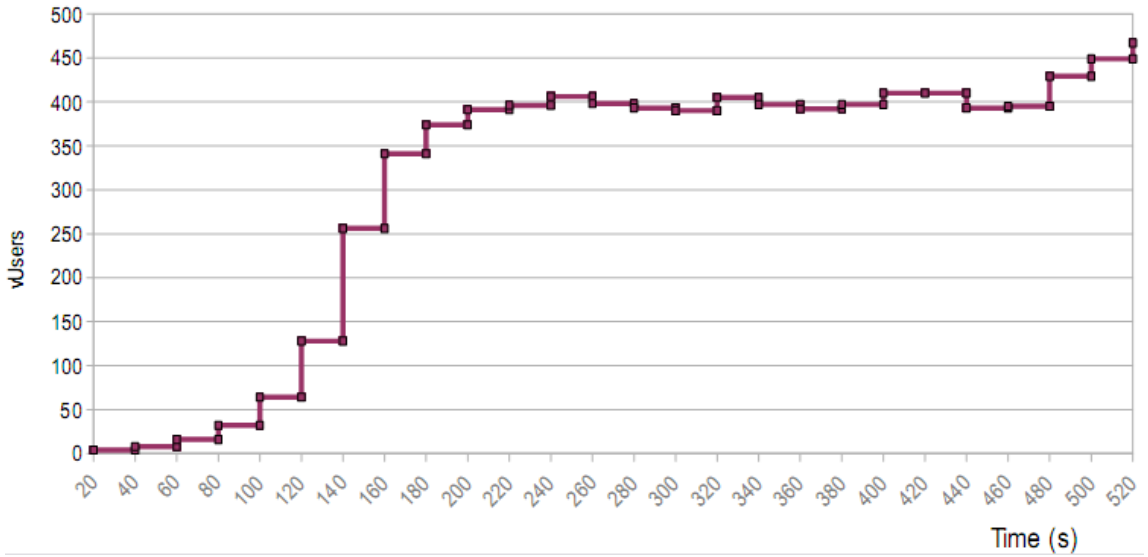




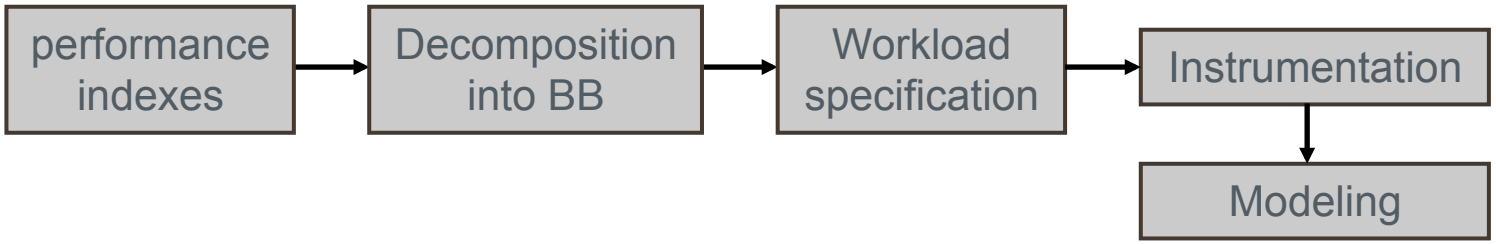
- After 3-4 minutes, A quick and good stabilization of the number of virtual users around 400 and an ESB load around 80%.

A self-regulated load injection experiment (2)





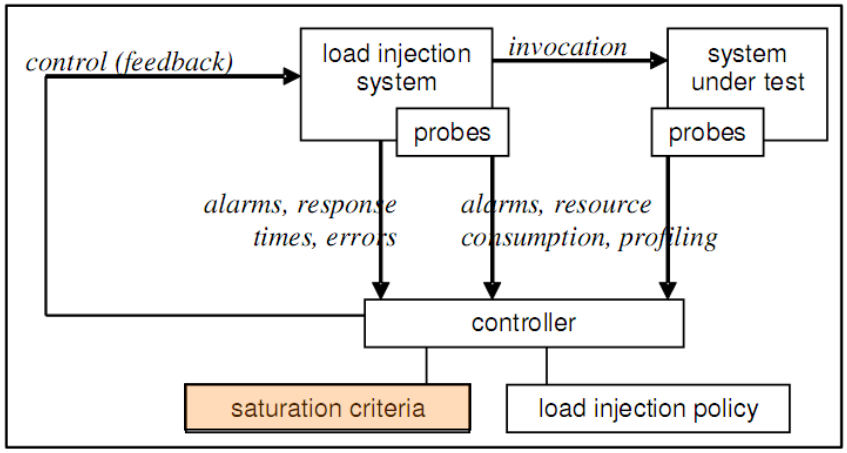
Conclusion



- We present methodology to characterize the performance of the black boxes
- A generic load injection framework based on CLIF to search saturation load and autonomously measure black boxes performance. (Experiment: Enterprise Service Bus).

Future work

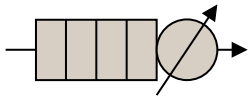
- Modelize interaction of the black boxes
- Saturation & stability criteria
- Simulation



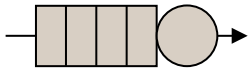
Thank you for your attention

Modeling :

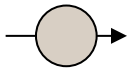
- Load-dependent black boxes : Queuing and service times depend on the load.



- Load-independent black boxes : The service time does not depend on the load.

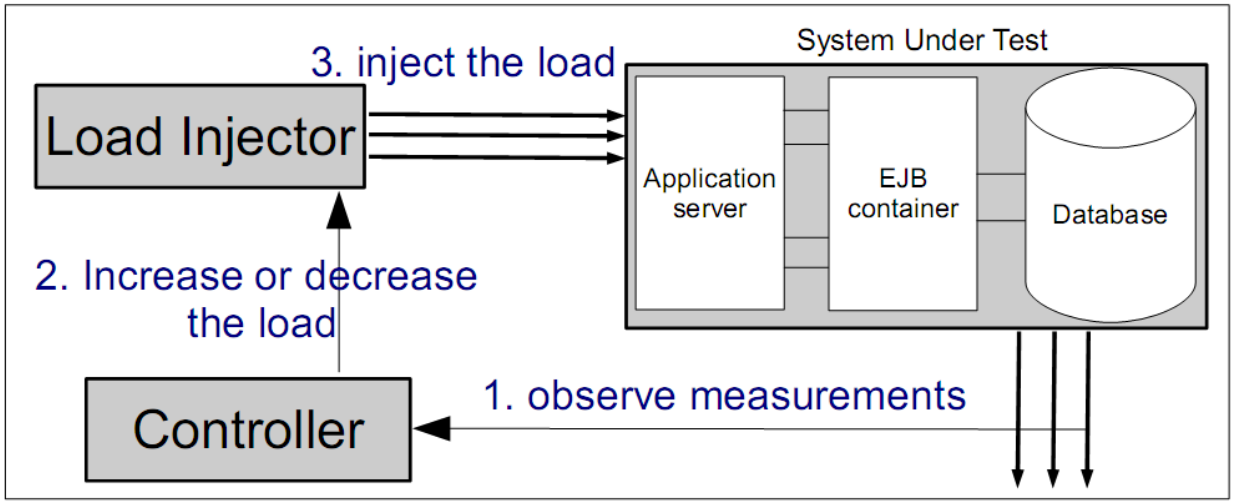


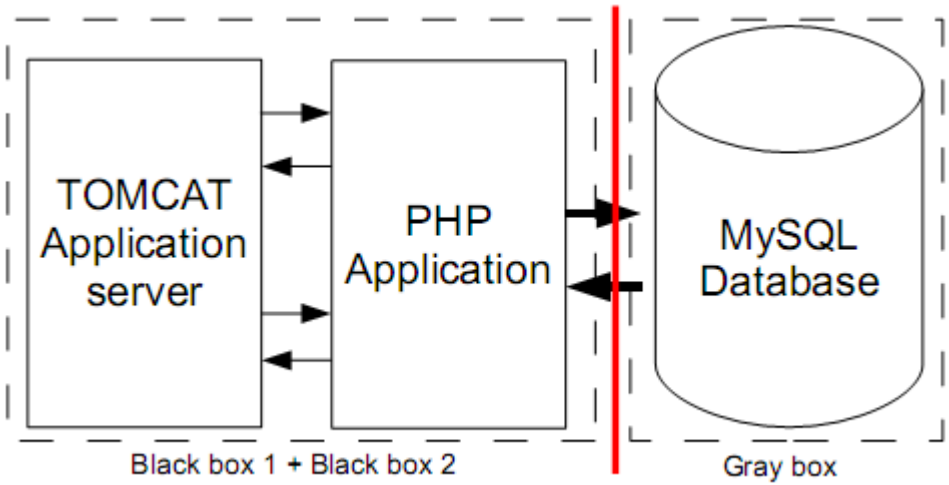
- Constant delay black boxes : Service time does not depend on the load and there is no queuing

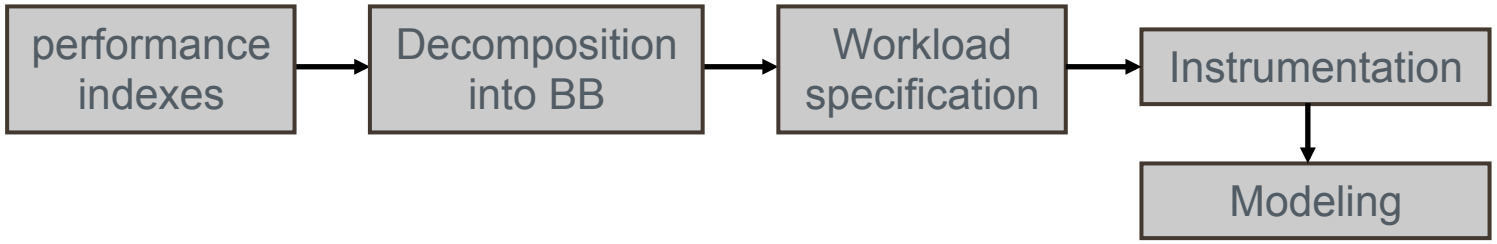


We define the type of each black box according to the test results

Saturation and Stability




TOMCAT
Application
server
PHP
Application
MySQL
Database
Black box 1 + Black box 2
Gray box


performance indexes
Decomposition into BB
Workload specification
Instrumentation
Modeling