HYBRID ADAPTIVE CHECKPOINTING FOR VIRTUAL MACHINE FAULT TOLERANCE

Abel Souza (Umeå University)
Alessandro Papadopoulos (Mälardalen University)
Luis Tomás (Red Hat)
David Gilbert (Red Hat)
Johan Tordsson (Umeå University)

abel.souza@umu.se

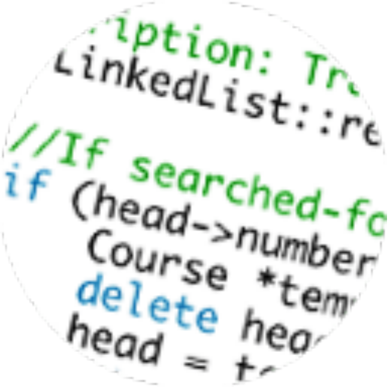
UMEÅ UNIVERSITY

INTRODUCTION

MOTIVATION

CURRENT SOLUTIONS & TRADEOFFS

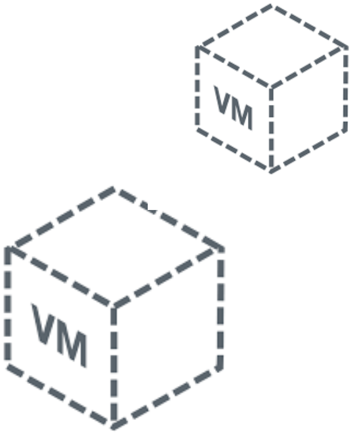
(i)



```
iption: Tr
LinkedList::re
//If searched-fo
if (head->number
Course *tem
delete hea
head = t
```

(ii)

(iii)



	Type	Feasibility*	Cost*	Overhead* **
(i)	Application	+-	$$$	>= 0
(ii)	Hardware	+	$$$	~ 0
(iii)	Agnostic (VMs)	+	$	> 0

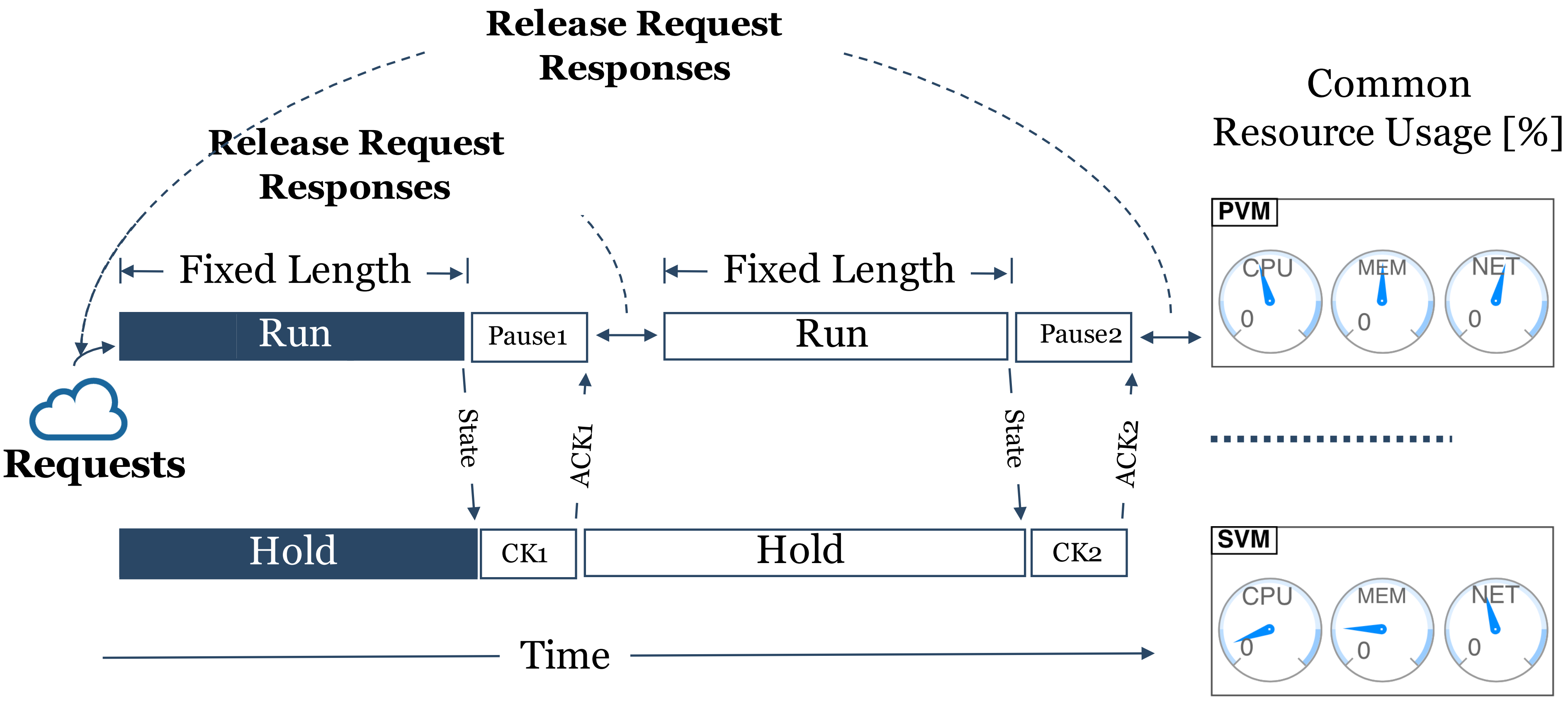
All solutions hide system errors from end-users, who can access **main** instance without interruptions even in the presence of faults.

* Educated Guess
** In Application Performance

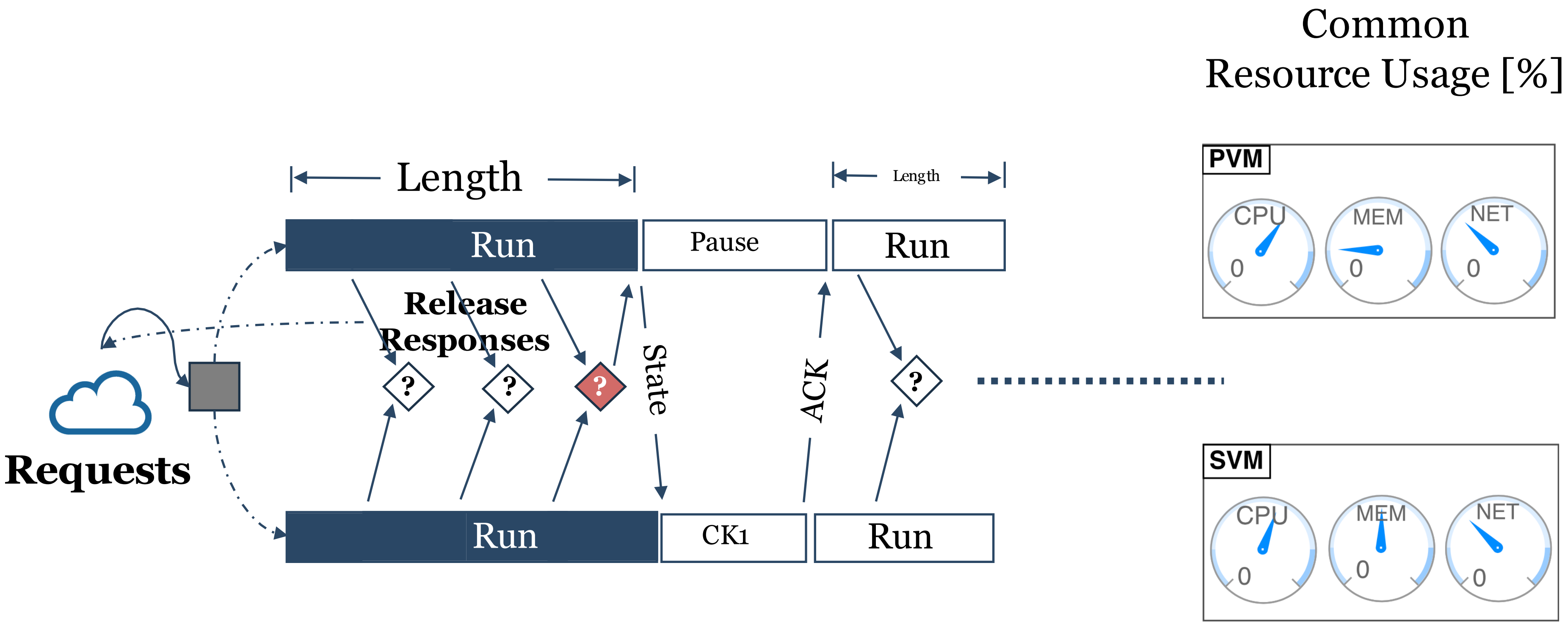
WEB/INTERACTIVE APPLICATIONS – AGNOSTIC FAULT TOLERANT APPROACHES

Active/Passive (A/P) vs Active/Active (A/A)

TRADITIONAL CHECKPOINT MODE (A/P)



<u>COLO</u> - <u>CO</u>ARSE GRAIN <u>LO</u>CK STEPPING* (A/A)



Common
Resource Usage [%]

Length

Length

Run

Pause

Run

Release
Responses

State

ACK

Requests

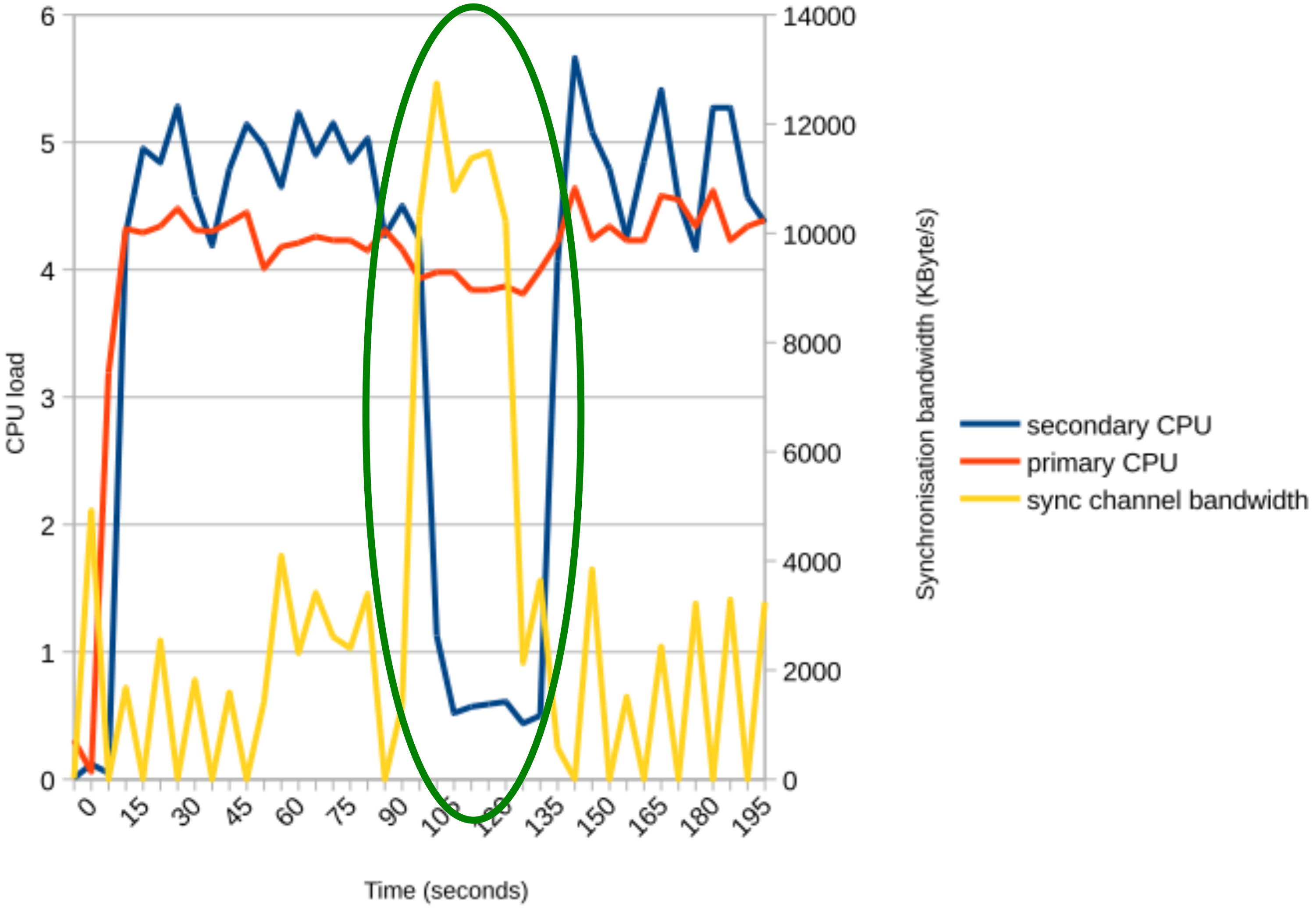
Run

CK1

Run

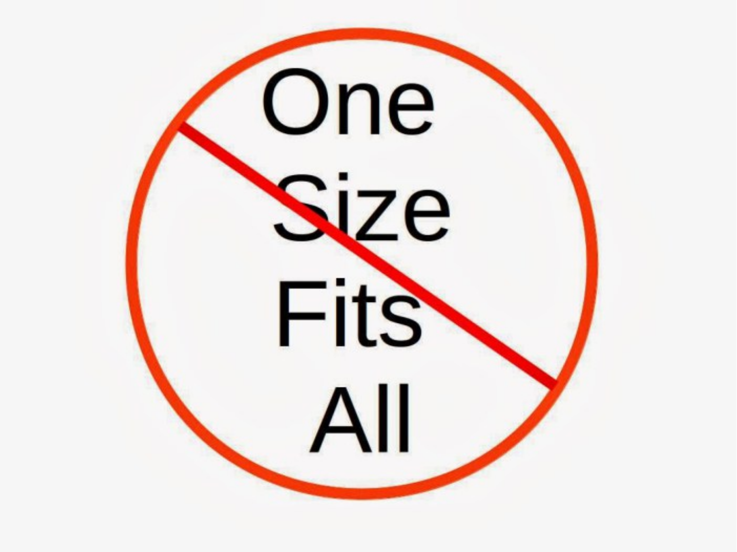
* Dong, YaoZu, et al. "COLO: COarse-grained LOck-stepping virtual machines for non-stop service" – ACM Symposium on Cloud Computing (2013)

RESOURCE TRADEOFFS: COLO & CHECKPOINT MODE



PROBLEM

COLO	CHECKPOINT
+ Lower latency - Workload can increase synchronization frequency: more **network usage** - More CPU usage	+ Less CPU usage - Greater latency - More network usage



IDEA: HYBRID APPROACH

- Main idea is to control how long we stay in COLO or checkpoint modes based on the frequency of checkpoints;

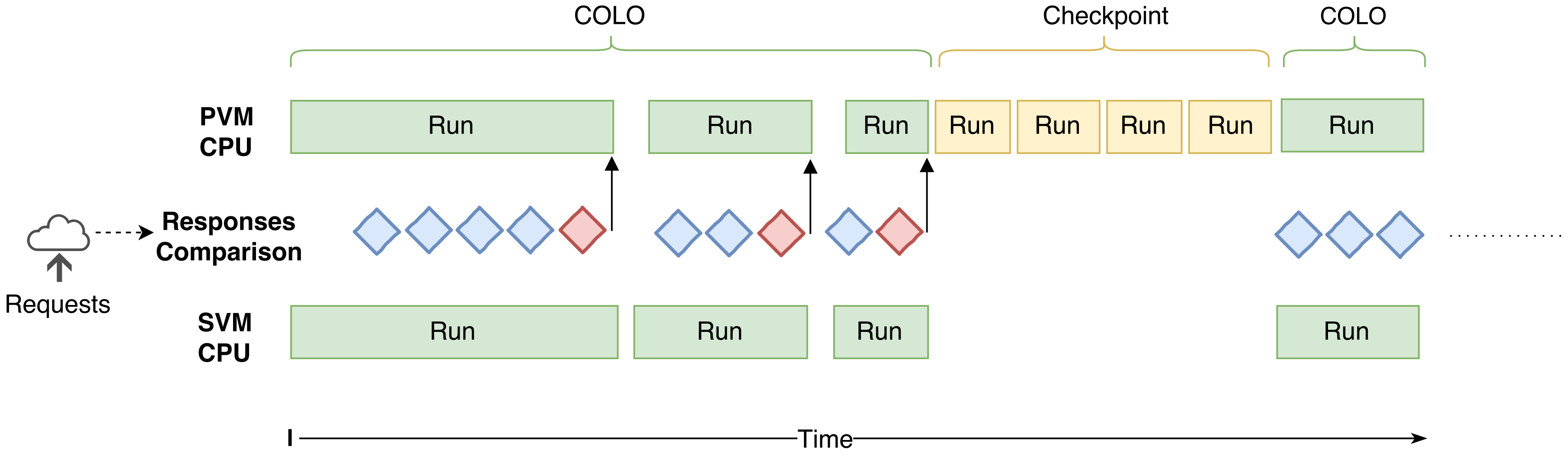
- This frequency indicates workload aspects to decrease VM Downtime Ratio, i.e., the overall amount of time the VM is paused due to a checkpoint;

Time **Not running**
VM

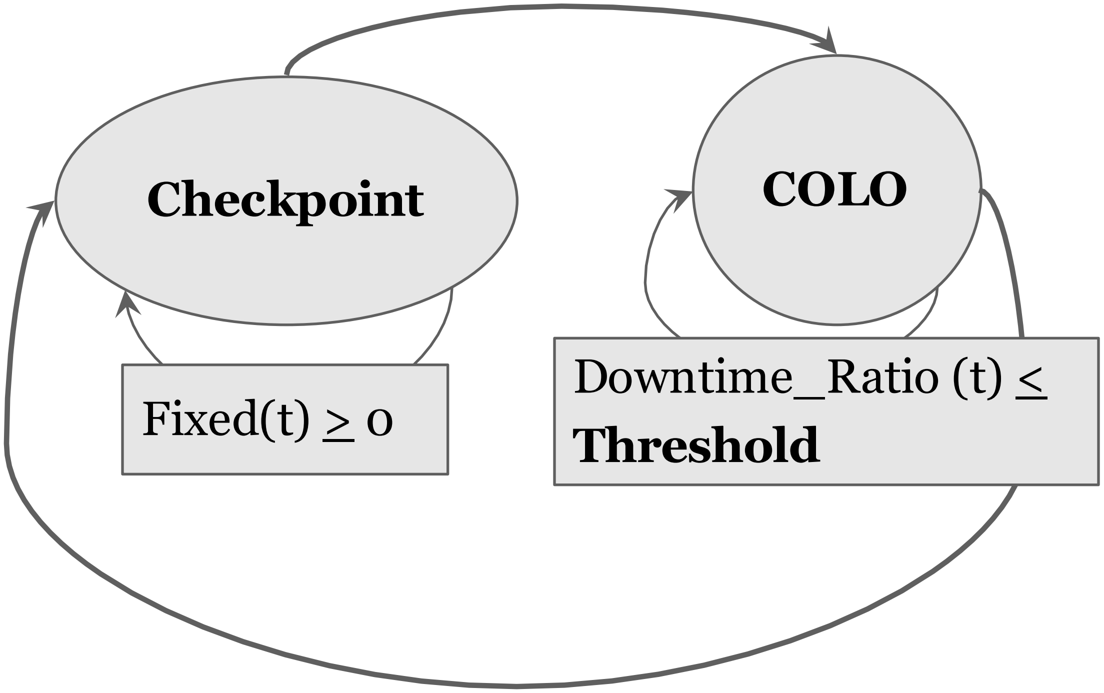
$$\textit{Downtime Ratio} = \frac{\text{Avg(Downtime)}}{\text{Avg(Uptime)}}$$

Time **Running**
VM

HYBRID APPROACH



HYBRID THRESHOLD MODE DESIGN



Checkpoint

COLO

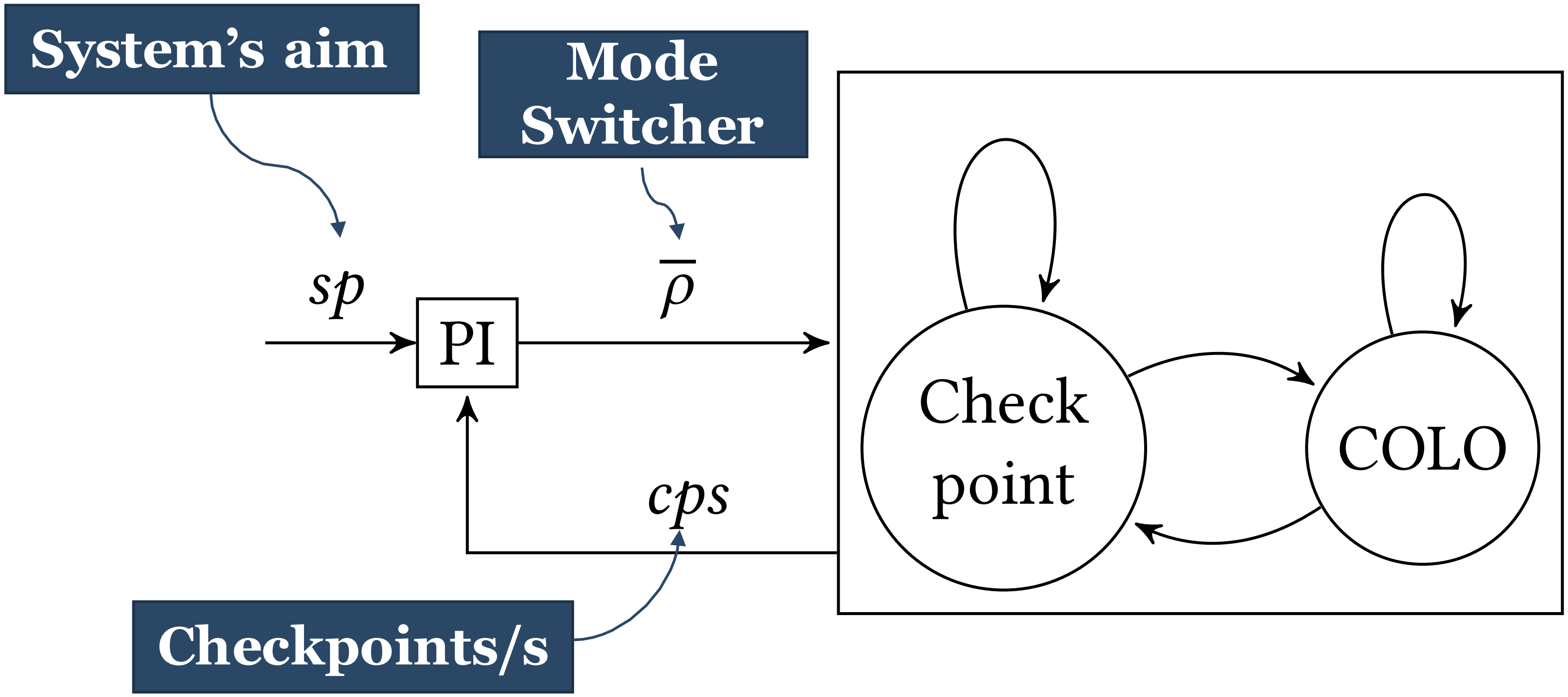
Fixed(t) ≥ 0

Downtime_Ratio (t) ≤ **Threshold**

If COLO is **good (on average)**, we **decrease** how long we will spend in Checkpoint mode by a BETA factor (for the next switching)

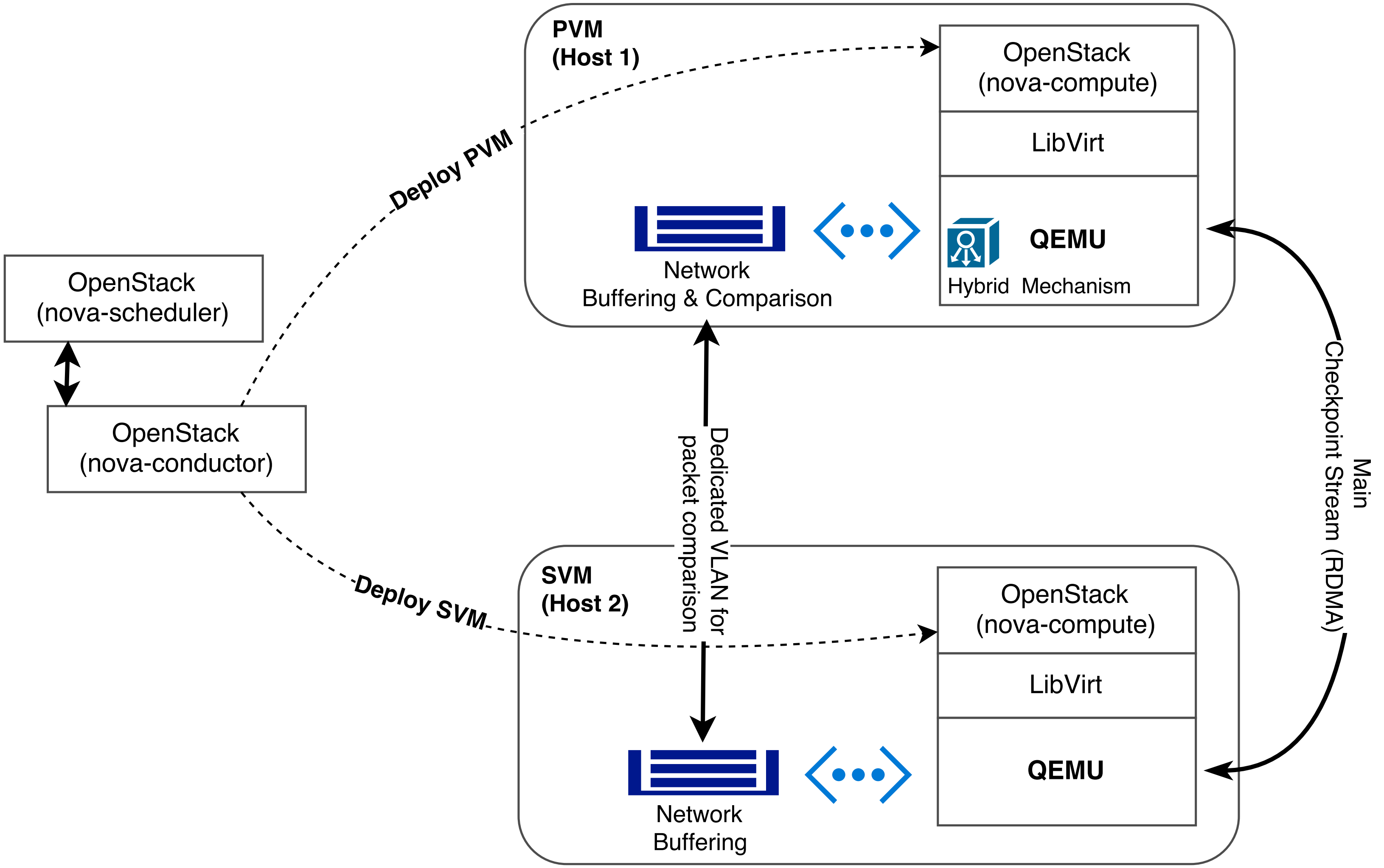
If COLO is **bad (on average)**, we **increase** how long we will spend in checkpoint mode for the next switching by an ALPHA factor (for the next switching)

HYBRID PROPORTIONAL-INTEGRAL CONTROLLER DESIGN



The PI-controller accepts a user-defined set-point (***sp***, system's aim) and ***cps*** (checkpoints per second) as inputs, both used to calculate ***ρ***, the controller's mode switcher.

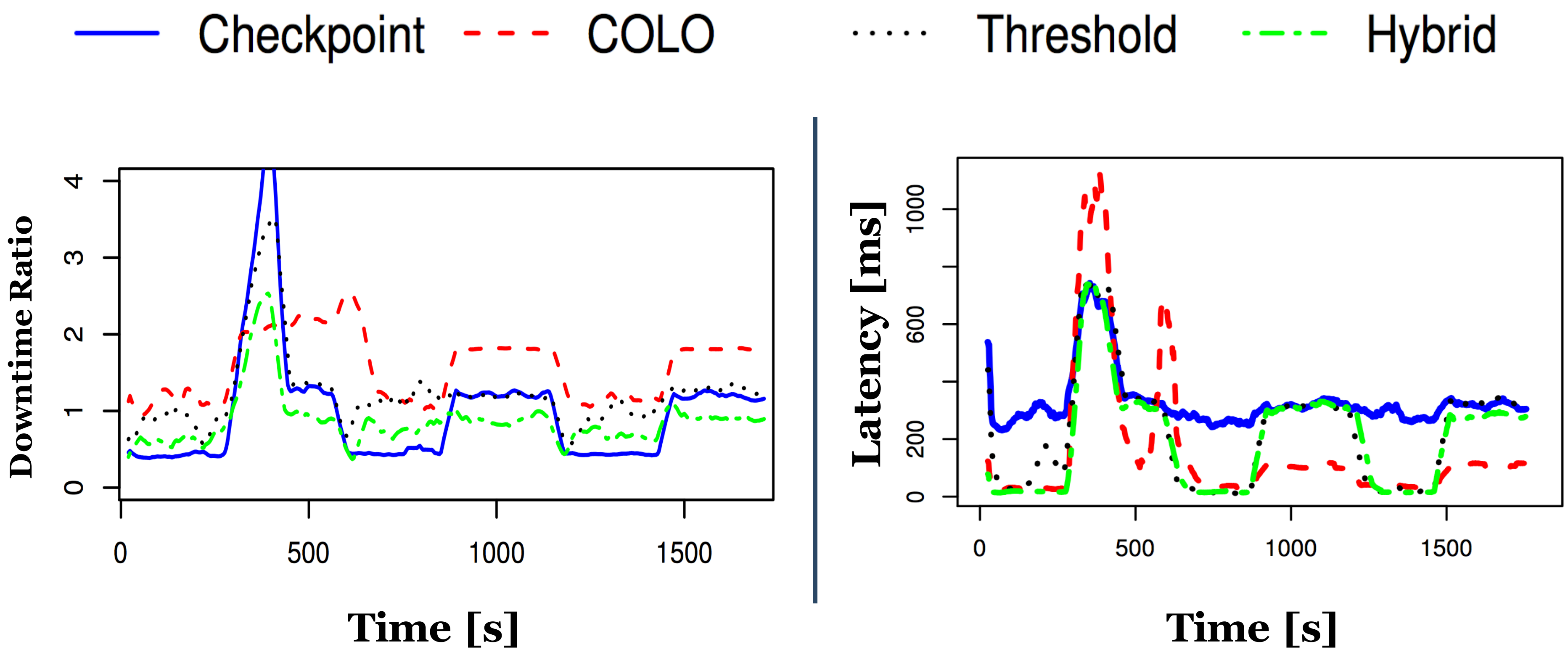
IMPLEMENTATION W/ OPENSTACK



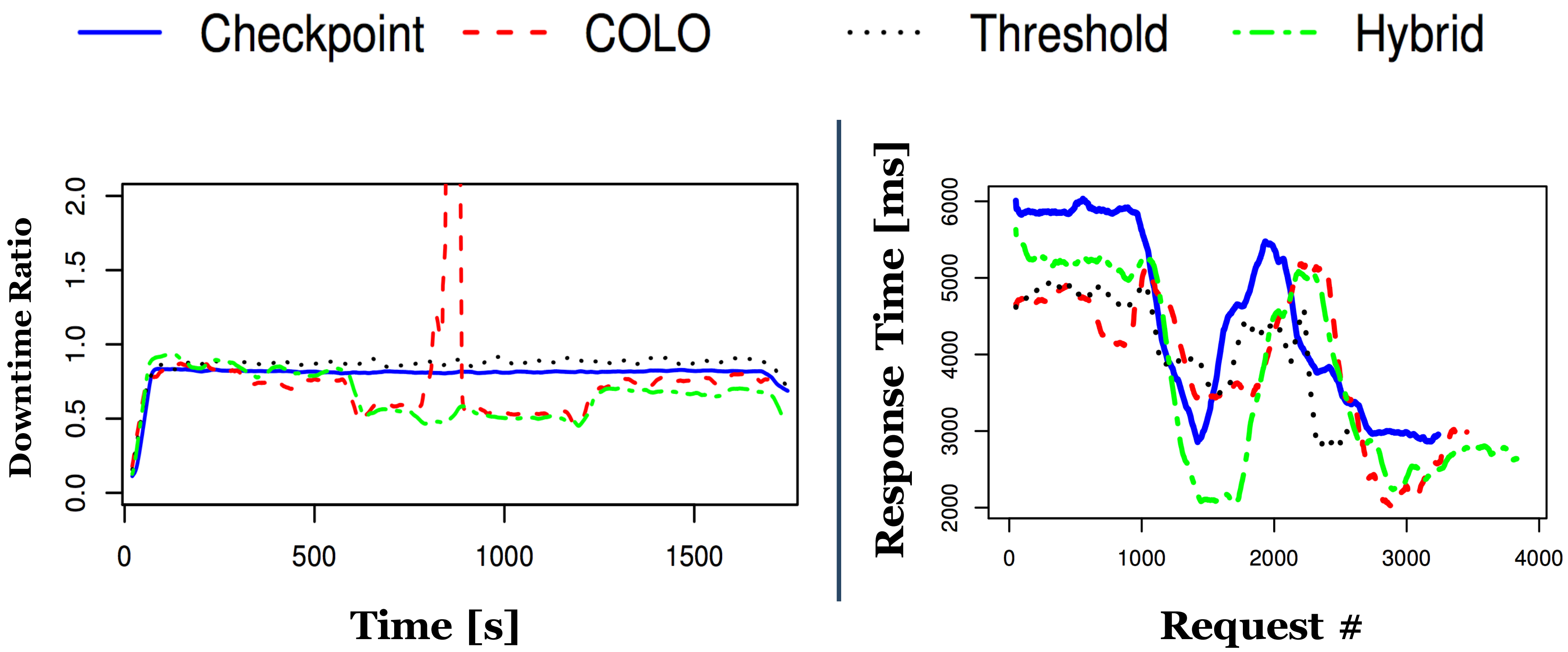
EVALUATION

- **Three real applications** were used in order to evaluate the proposed hybrid approaches:
  - RUBiS online auction benchmark;
    - Database (I/O);
    - **Very deterministic**;
  - BugZilla Tracking System;
    - Multi-threaded Bug filing application;
    - **Non-deterministic**;
  - Video Streaming.
    - Heavily-threaded application;
    - **Highly non-deterministic**;
- All applications run 10 times, 30 minutes in each mode:
  - Checkpoint, COLO, Threshold and Controlled (Hybrid).

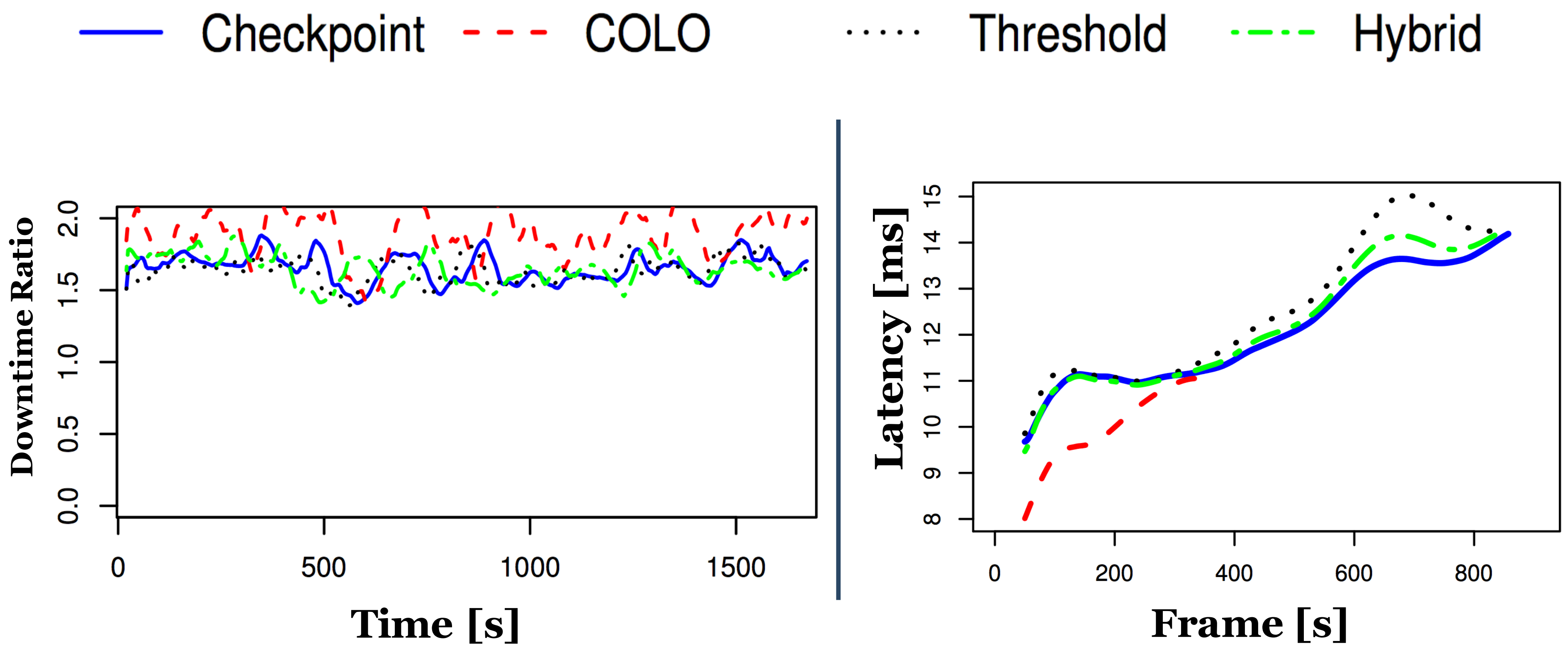
RESULTS - SYNTHETIC WORKLOAD



RESULTS – BUGZILLA WORKLOAD



RESULTS – VIDEO STREAMING WORKLOAD



DISCUSSION

- Average latency and throughput are where tradeoffs can be seen
- Our solution has not under-performed in any scenario, only by overheads for CPU and Network usages
- In particular, unpredictable behavior may be caused by many different factors:
    - Timestamp, unique-IDs per request, multi-threating…

CONCLUSIONS

- Controller follows workload behavior and decreases Downtime Ratios for each VM, without impact in application performance

- Machine/Reinforcement Learning could be used to learn workload

    - Potential to speed up the mode learning/convergence process;

    - Additional metrics for decision making policy (such as Performance)

- The adaptation ability is relevant in many use-cases

    - Operators offer infrastructure without asking what users will run.

www.orbitproject.eu

HYBRID ADAPTIVE CHECKPOINTING FOR VIRTUAL MACHINE FAULT TOLERANCE

Abel Souza (Umeå University, *Sweden*)
Alessandro Papadopoulos (Mälardalen University, *Sweden*)
Luis Tomás (Red Hat, *Spain*)
David Gilbert (Red Hat, *U.K.*)
Johan Tordsson (Umeå University, *Sweden*)

abel.souza@umu.se

UMEÅ UNIVERSITY

The research leading to these results has received funding from the EC Seventh Framework Programme FP7/2007-2011 under grant agreement n° 609828