Identification of availability and performance bottlenecks in cloud computing systems:

An approach based on hierarchical models and sensitivity analysis

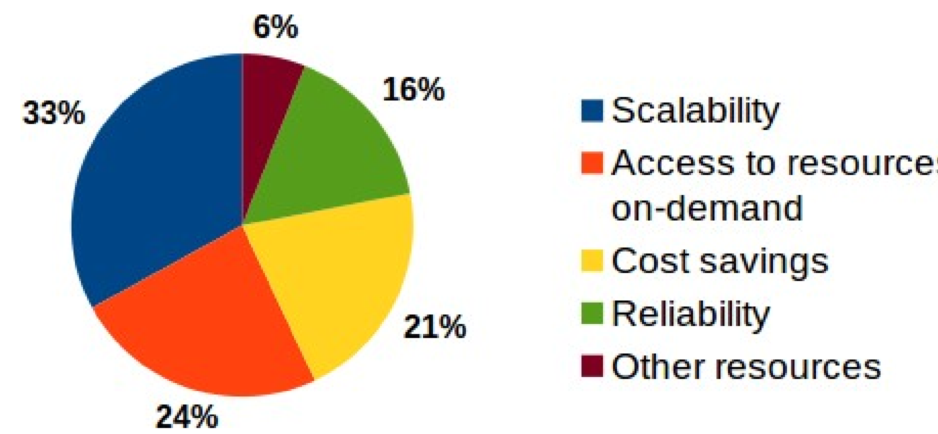
Rubens de Souza Matos Júnior

Advisor: Prof. Paulo Maciel

Universidade Federal de Pernambuco

Motivation

- Among the major reasons mentioned for **adoption of cloud computing** are:
  - Scalability
  - Access to resources on-demand
  - Cost savings
  - Reliability
- **Problems** in large cloud providers show the importance of proper availability and performance **planning** for cloud infrastructures and their hosted services.



Source: rightscale.com

Home / Reviews / Software / Security / **Amazon Cloud Outage Hits Netflix, Foursquare**

Amazon Cloud Outage Hits Netflix, Foursquare

BY CHLOE ALBANESIUS AUGUST 9, 2011 11:14AM EST 6 COMMENTS

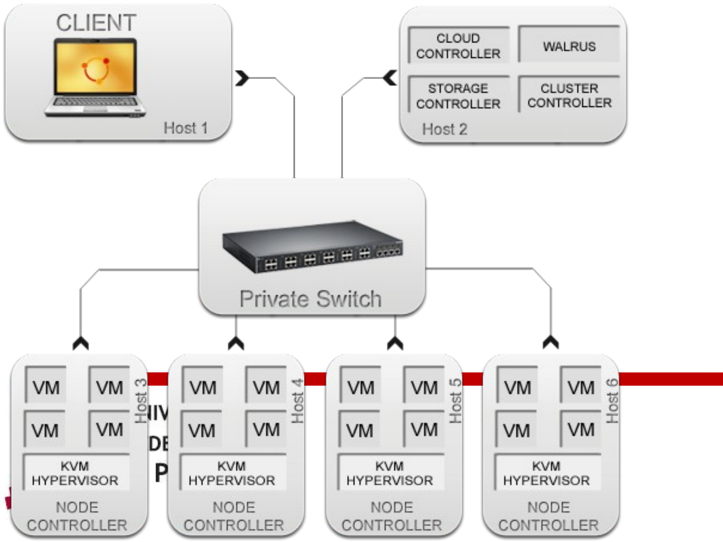
In the same week that a lightning strike in Dublin knocked out service for some European users of A Microsoft's cloud services, Amazon also suffered a stateside cloud outage that affected popular serv Foursquare, Reddit, and Netflix.

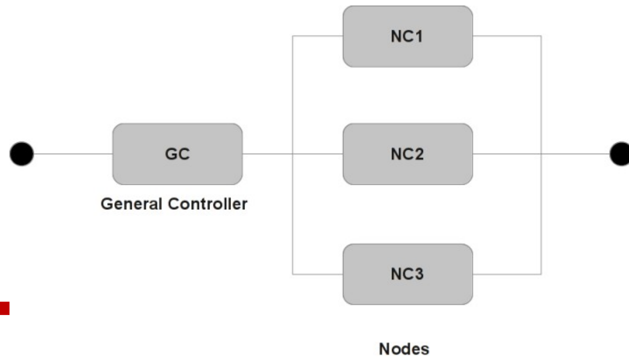
Universidade Federal de Pernambuco

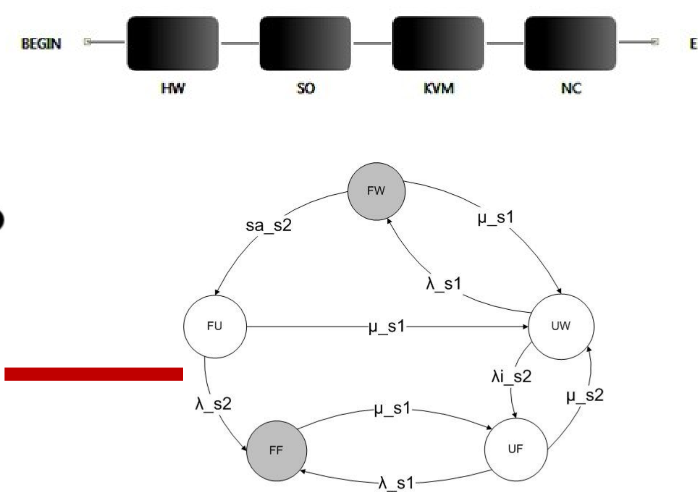
Motivation

- How to evaluate the performance and availability of cloud computing systems, and **detect bottlenecks** to propose improvements?

- Cloud computing systems have great complexity
  - Many hardware and software components
  - Interdependence between components
  - **Solution: Hierarchical modeling**





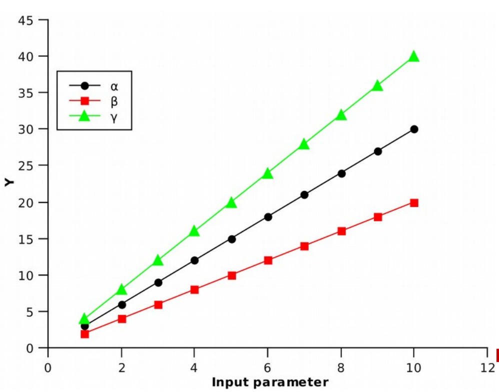


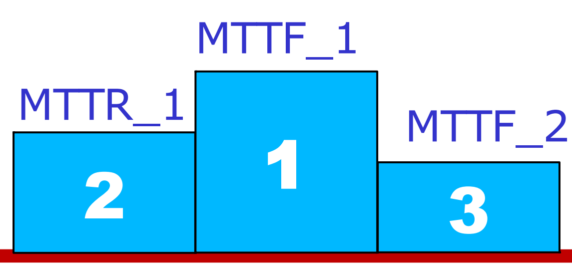
Motivation

Centro de Informática U·F·P·E

- How to evaluate the performance and availability of cloud computing systems, and **detect bottlenecks** to propose improvements?

- How to identify what will bring the biggest **gain in quality of service**?
  - More powerful and reliable hardware ?
  - More advanced architecture ?
  - A software that provides flexibility, autonomy, resilience ?
  - **Solution: Sensitivity analysis, adapted for hierarchical models**





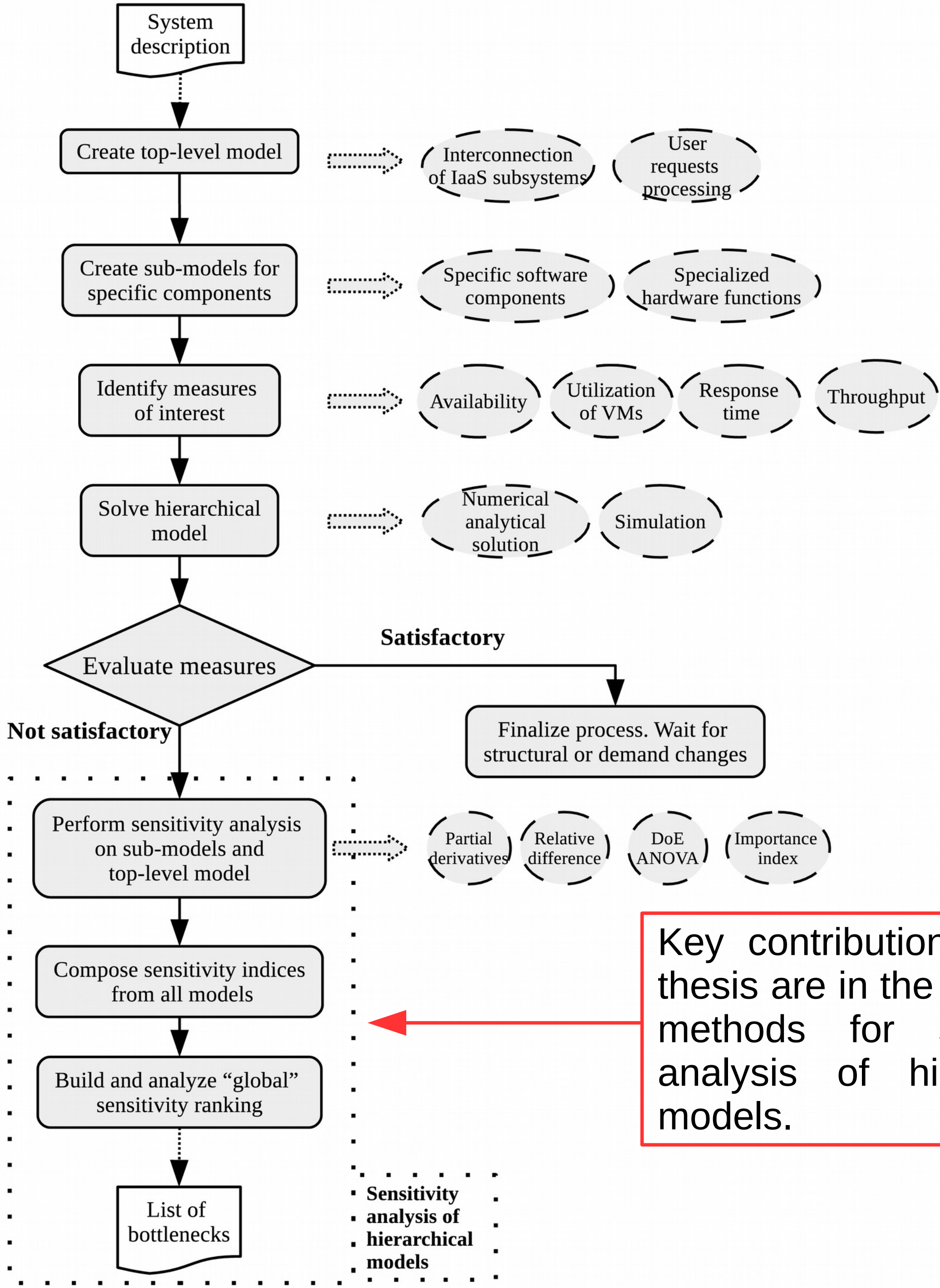
Parameter	S(Avail)
q	0.8
A	0.8
s	0.7
...	...
m	0.3
p	0.1

Unified ranking of parameters

Objectives

- The main objective of this thesis is to propose methods for detection of performance and availability bottlenecks in cloud computing systems.

Supporting methodology

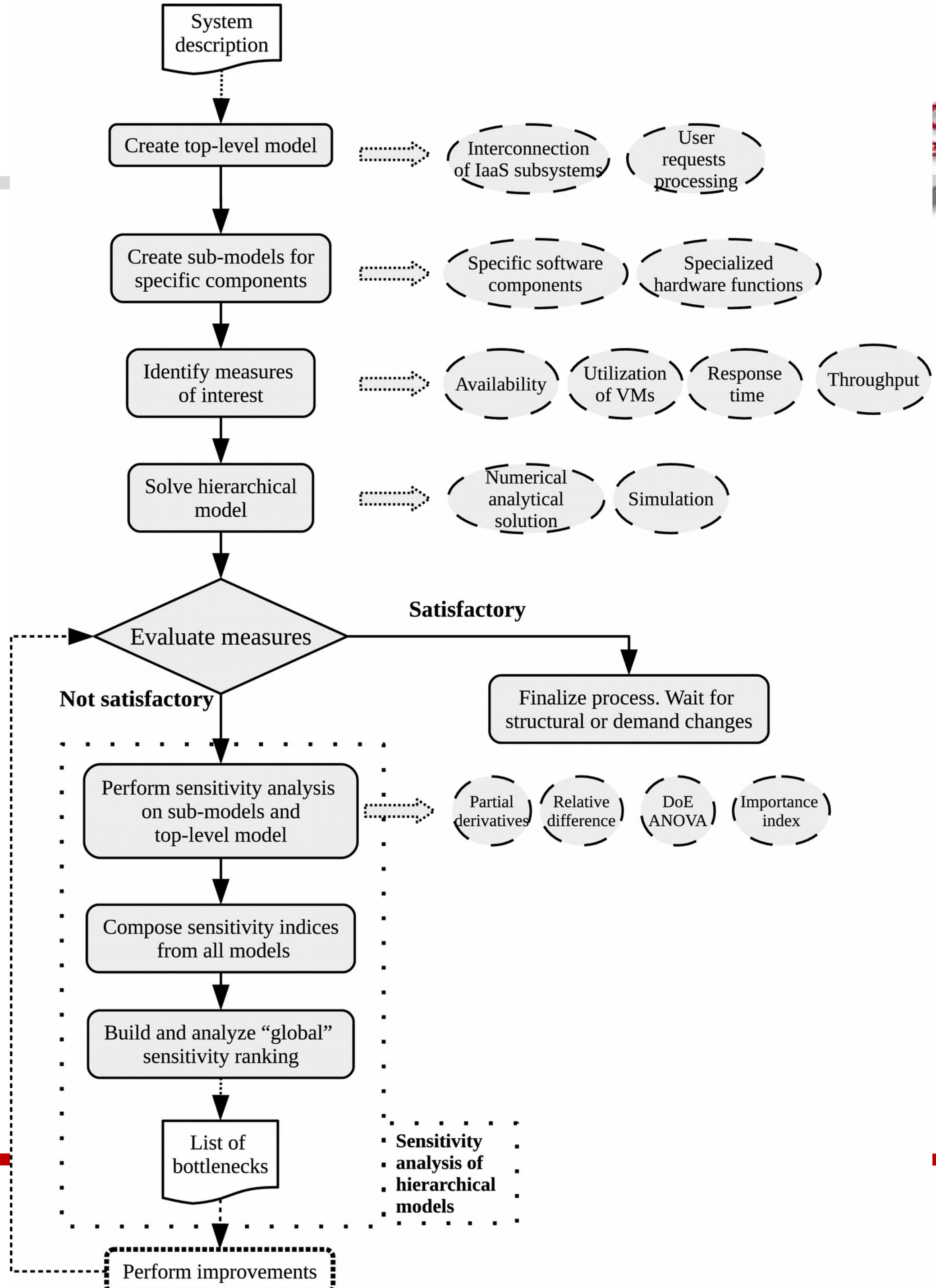


Key contributions of this thesis are in the proposed methods for sensitivity analysis of hierarchical models.

Universidade Federal de Pernambuco

Centro de Informática UFPE

Supporting methodology



Bottlenecks are the targets for potential improvements, transforming it in an iterative methodology

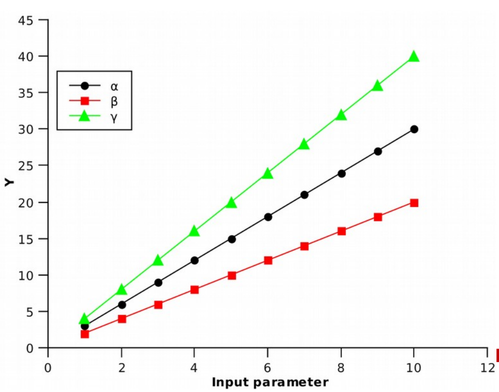
Centro de Informática U·F·P·E

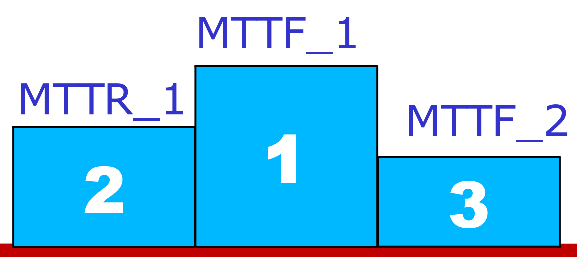
Universidade Federal de Pernambuco

Motivation

- How to evaluate the performance and availability of cloud computing systems, and **detect bottlenecks** to propose improvements?

- How to identify what will bring the biggest **gain in quality of service**?
  - More powerful and reliable hardware ?
  - More advanced architecture ?
  - A software that provides flexibility, autonomy, resilience ?
  - **Solution: Sensitivity analysis, adapted for hierarchical models**

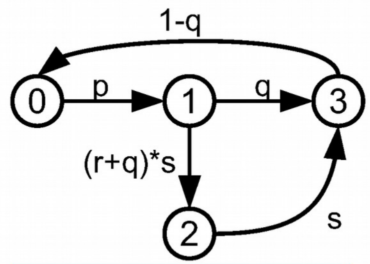


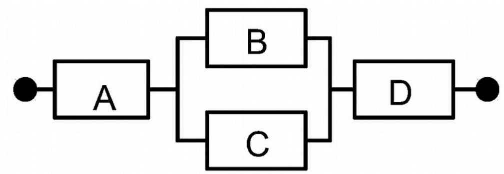


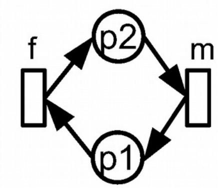
Parameter	S(Avail)
q	0.8
A	0.8
s	0.7
...	...
m	0.3
p	0.1

Unified ranking of parameters

Composition of sensitivity indices



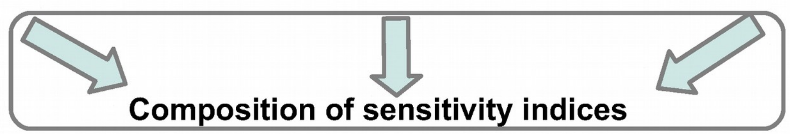




Parameter	S(B)
q	0.9
s	0.8
r	0.4
p	0.2

Parameter	S(Avail)
A	0.7
D	0.6
C	0.4
B	0.4

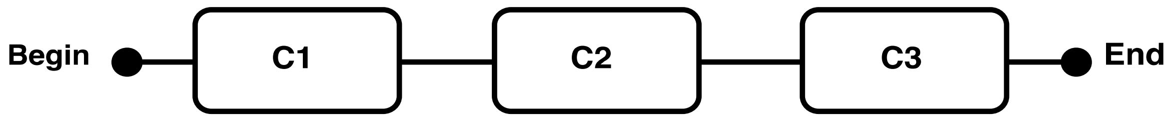
Parameter	S(C)
f	0.7
m	0.2



Parameter	S(Avail)
q	0.8
A	0.8
s	0.7
...	...
m	0.3
p	0.1

Unified ranking of parameters

Proposed composition techniques: RBD + CTMCs



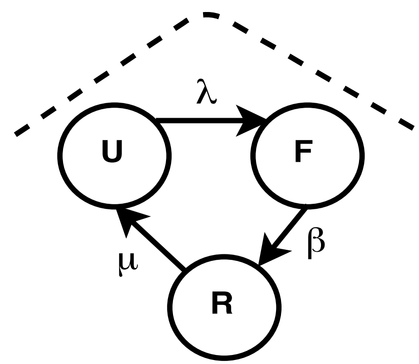
$A_{C1} = \frac{\mu}{\gamma + \mu}$ $A_{C2} = \frac{\mu}{\rho + \mu}$ $A_{C3} = \frac{\beta\,\mu}{\lambda\,\mu + \beta\,(\lambda + \mu)}$

Structural equation:

$A = A_{C1} \times A_{C2} \times A_{C3}$

Derivative structural equation:

$S_{\mu}(A) = S_{\mu}(A_{C1}) \times A_{C2} \times A_{C3} + A_{C1} \times S_{\mu}(A_{C2}) \times A_{C3} + A_{C1} \times A_{C2} \times S_{\mu}(A_{C3})$

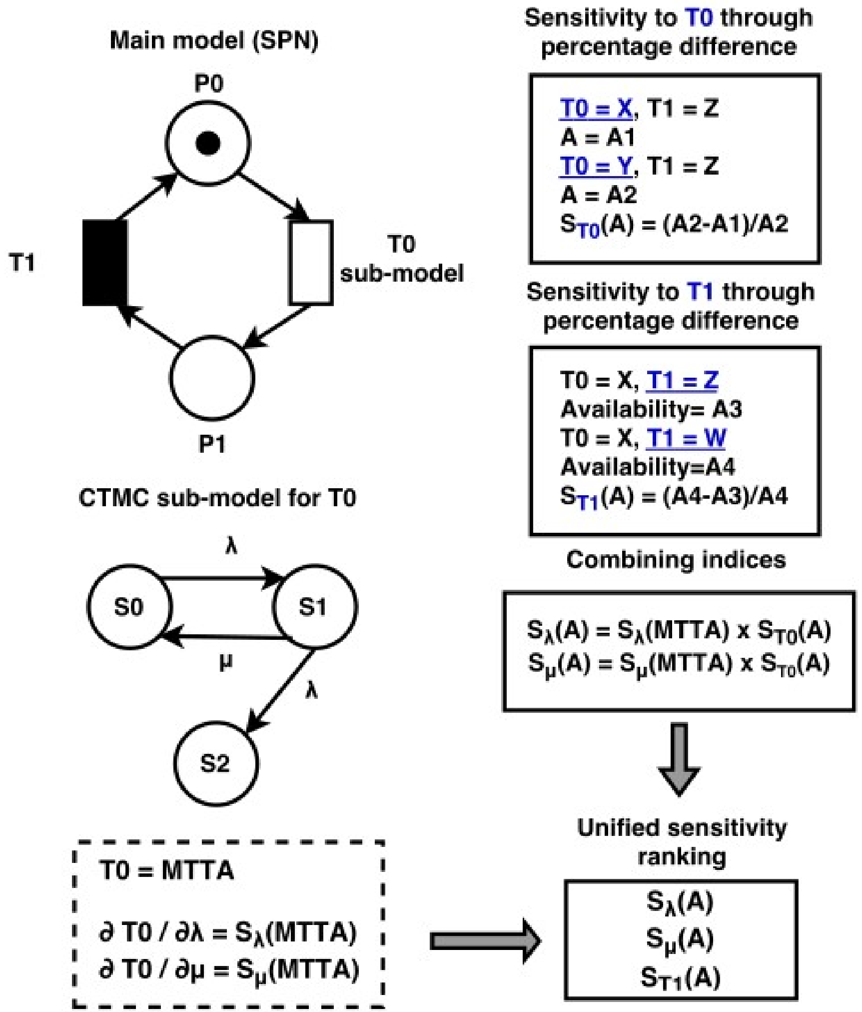


$S_{\mu}(A_{C1}) = \frac{\gamma}{(\gamma + \mu)^2}$ $S_{\mu}(A_{C2}) = \frac{\rho}{(\rho + \mu)^2}$ $S_{\mu}(A_{C3}) = \frac{\beta^2\,\lambda}{(\lambda\,\mu + \beta\,(\lambda + \mu))^2}$

In some cases, CTMC sub-models can be solved through closed-form equations. Their partial derivatives will provide the sensitivity indices.

UNIVERSIDADE FEDERAL DE PERNAMBUCO

Proposed composition techniques: SPN (simulation) + CTMCs

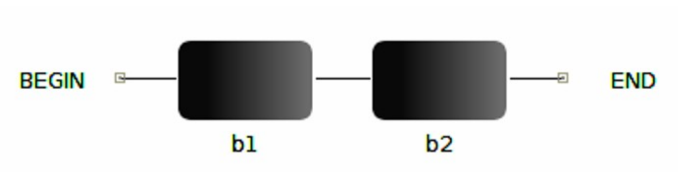


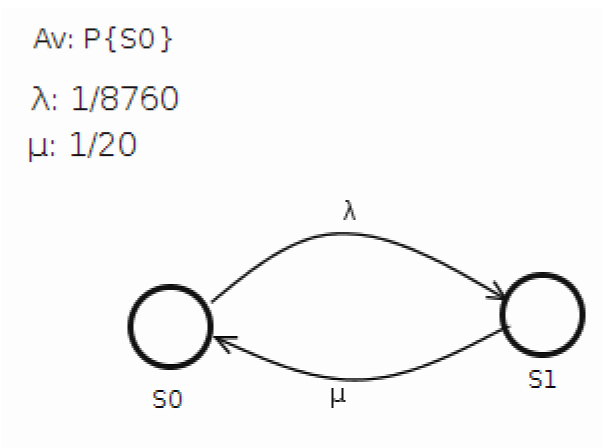
When the SPN can only be solved through **simulation**, the indices from CTMC sub-models are multiplied by indices of corresponding SPN transitions. Therefore, we follow the **chain rule**:

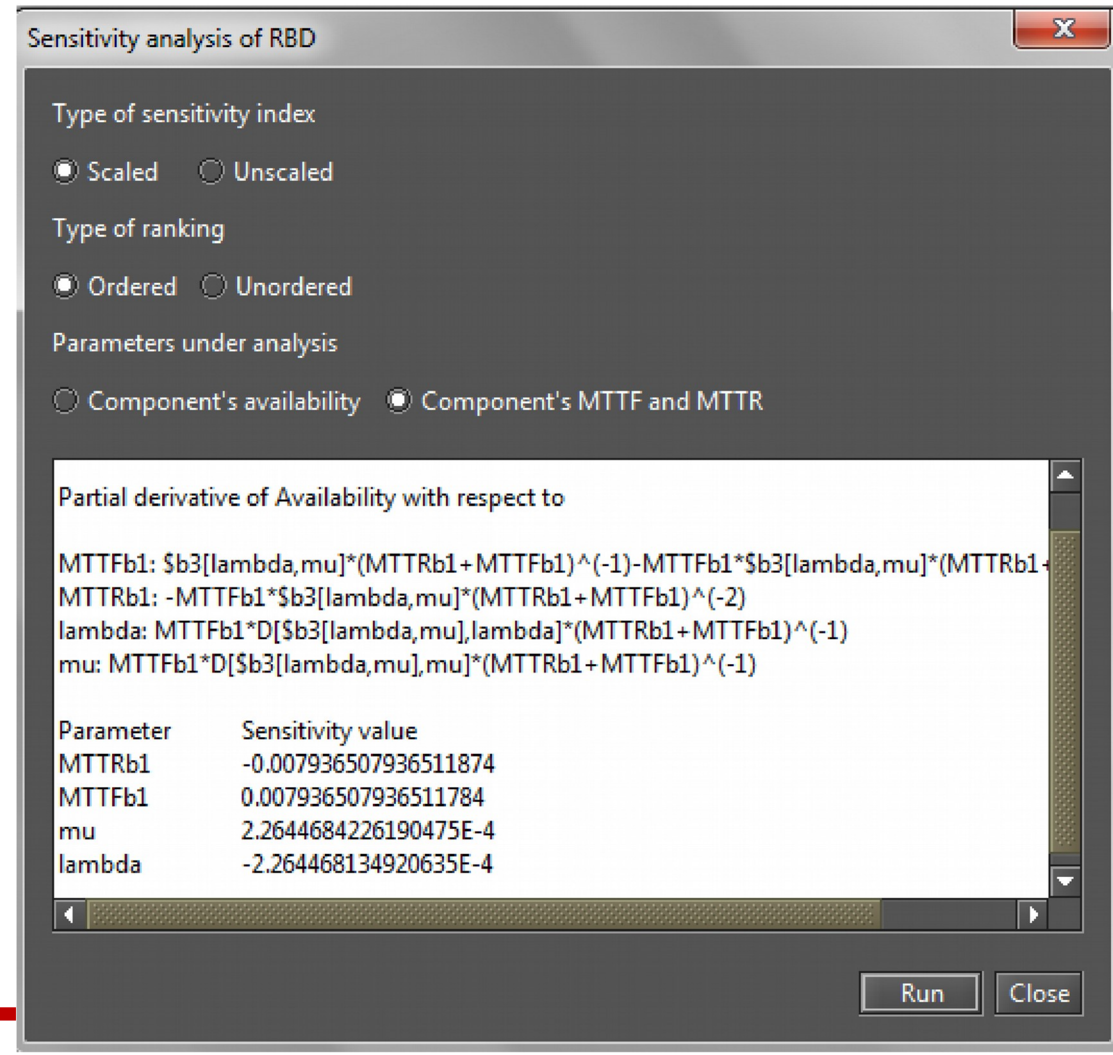
$$\frac{dz}{dx} = \frac{dz}{dy} \cdot \frac{dy}{dx},$$

z is the measure from SPN model
x is a parameter from CTMC sub-model
y is a transition from SPN model which has the delay as a function of the parameter x

Implementation on Mercury tool

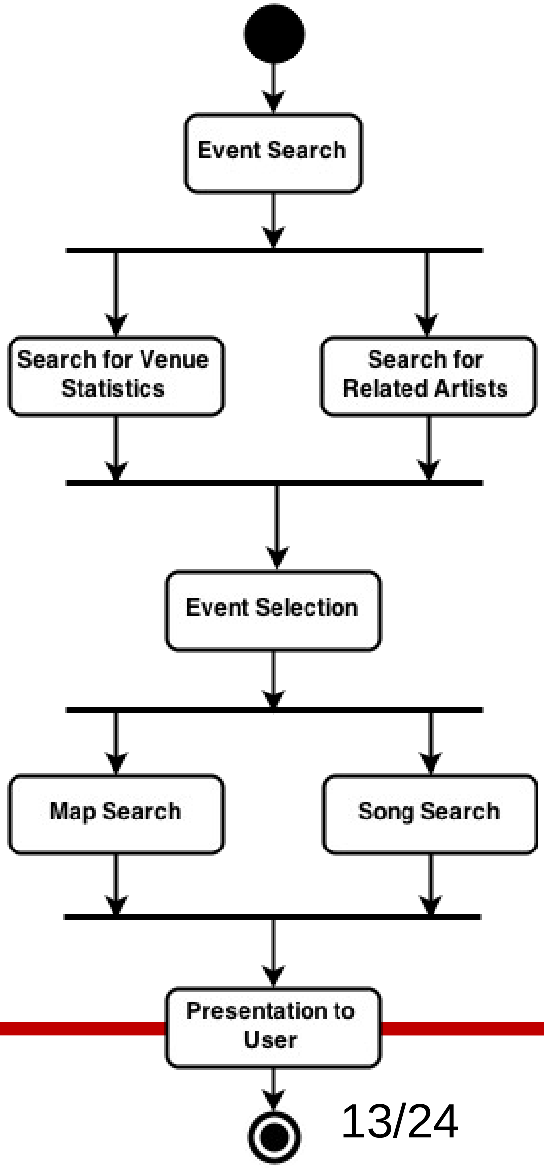






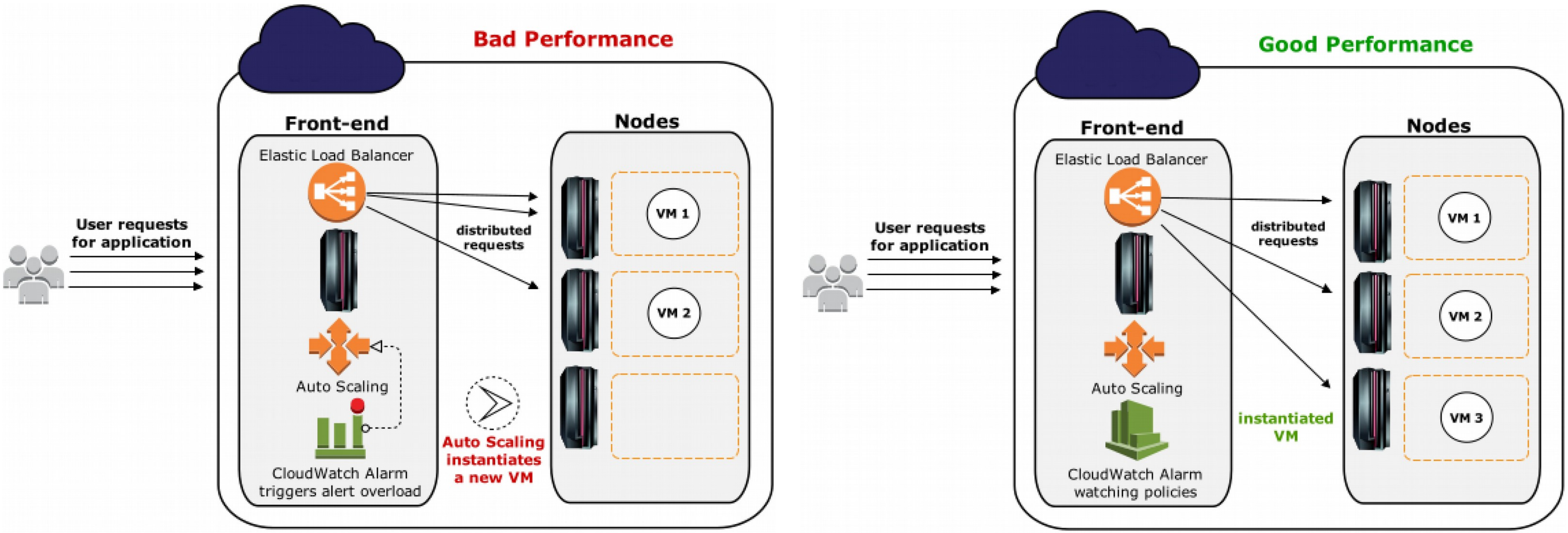
Case study: Composite web services on private cloud with autoscaling

- Composite **web services** for musical events recommendation
- This mashup runs on a private cloud, with elasticity resources: **automatic creation** and **termination of VMs** according to the workload



UNIVERSIDADE FEDERAL DE PERNAMBUCO

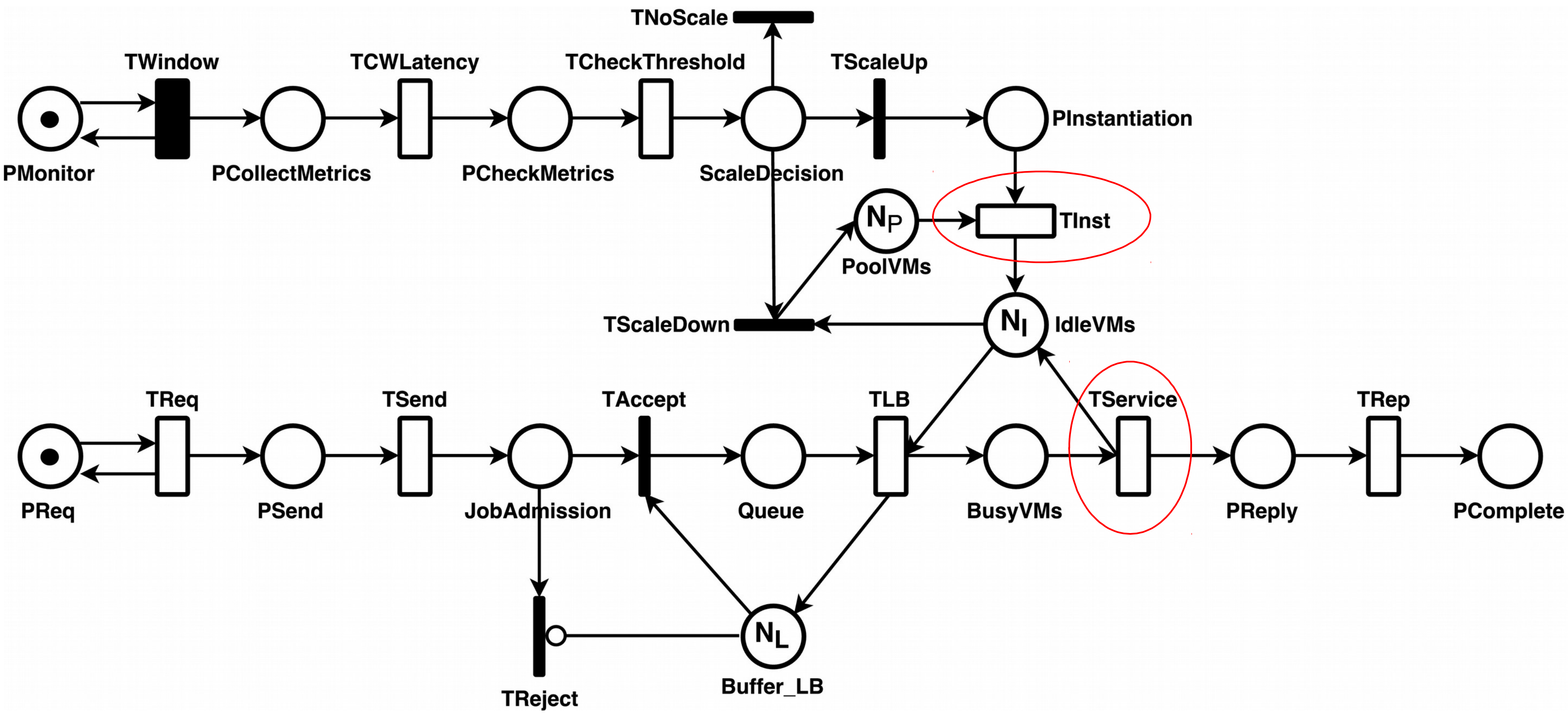
Composite web services on private cloud with autoscaling



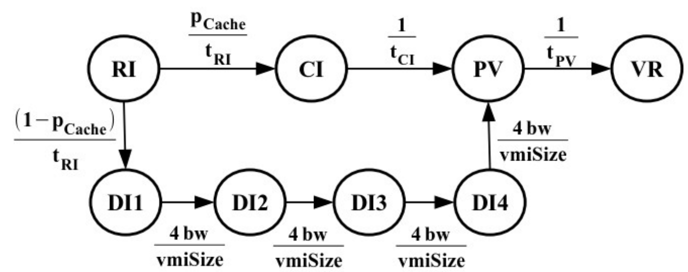
Composite web services on private cloud with autoscaling

- 3 models: 1 SPN + 2 CTMCs:
  - Workload / **autoscaling**
  - VM **instantiation**
  - Web service **execution**

Composite web services on private cloud with autoscaling (Step 1)

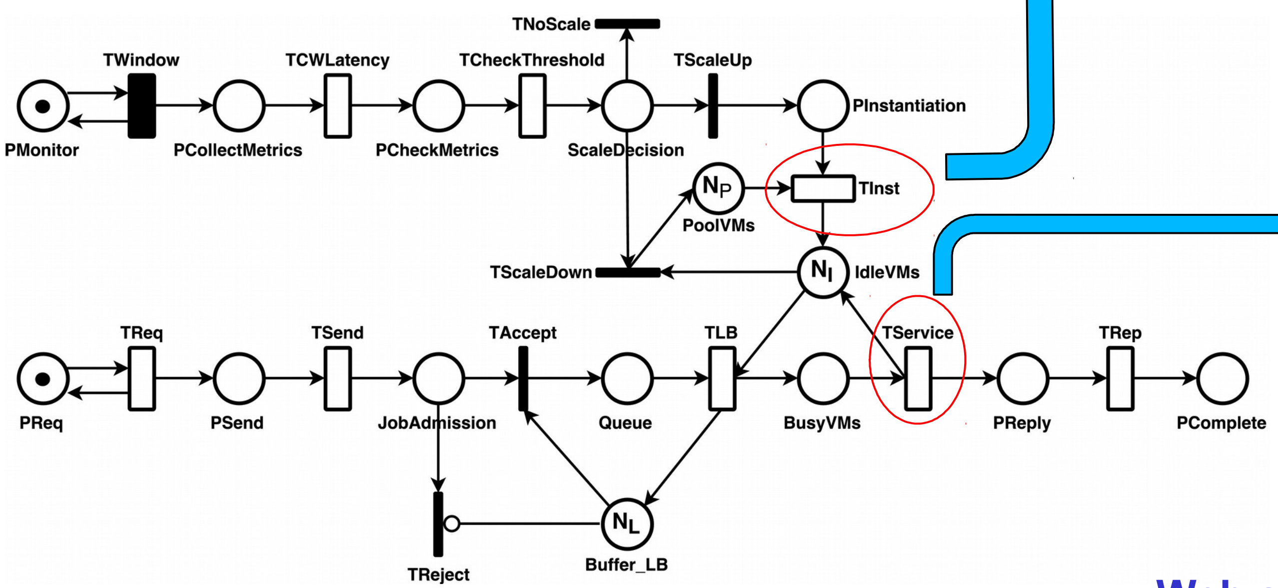


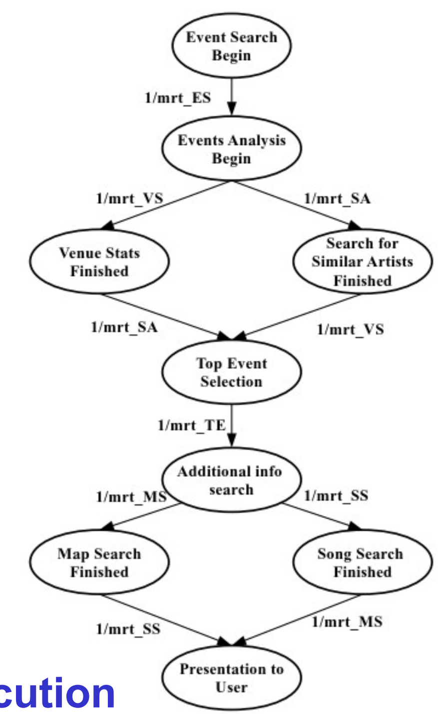
Composite web services on private cloud with autoscaling (Step 2)



VM instantiation

System representation

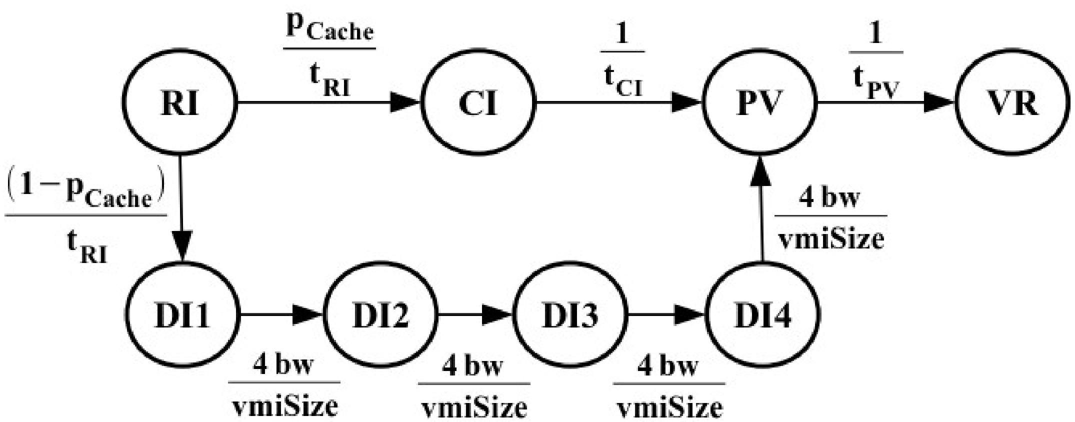


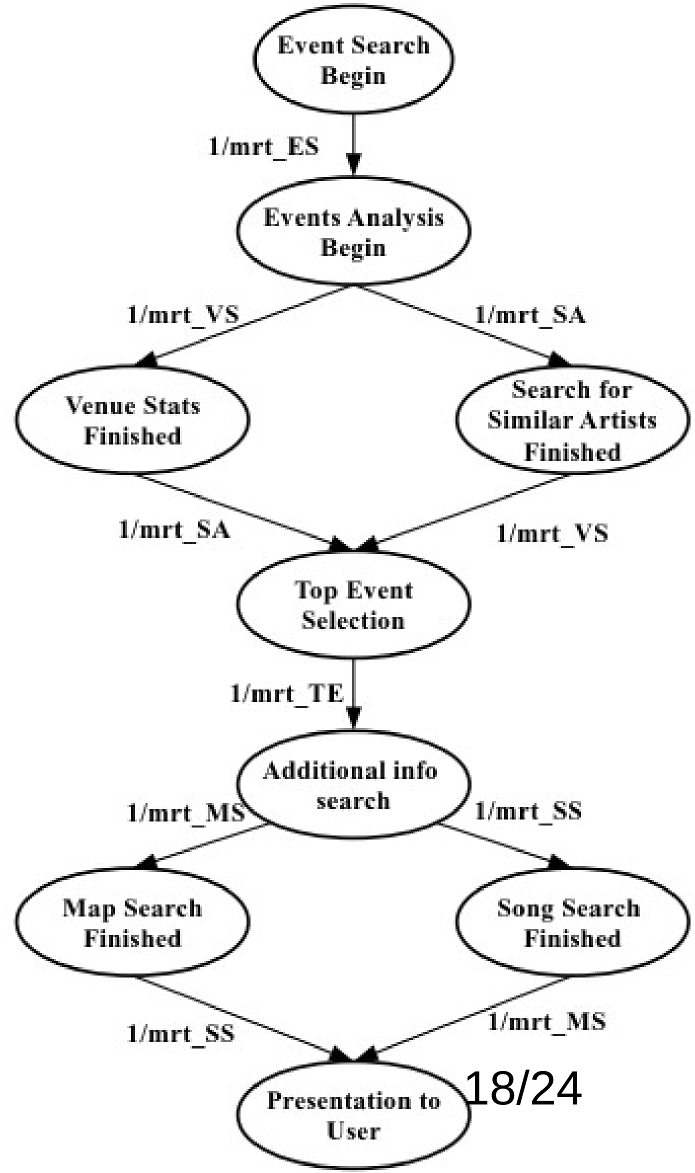


Web service execution

UNIVERSIDADE FEDERAL DE PERNAMBUCO

Composite web services on private cloud with autoscaling (Step 2)





Composite web services on private cloud with autoscaling (Step 3, 4, and 5)

Centro de Informática U·F·P·E

Performance measures

Measure	Expression	Value
Utilization of VMs (%)	E{#BusyVMs})/(E{#IdleVMs}+E{#BusyVMs})	38.3 %
Average number of busy VMs	E{#BusyVMs}	1.716
Average number of idle VMs	E{#IdleVMs}	2.773
LB queue size (#of requests)	E{#Queue}	0.432
Mean response time - Rsp - (s)	NRequests/(P{#PReply>0}×(1/TReply))	9.029 s

This is the metric of most interest for the user and it is not is a satisfactory level.

Composite web services on private cloud with autoscaling (Steps 6 and 7)

Sensitivity ranking for the main model

Parameter	S(Rsp)
TService	0.45763
TLB	0.13788
TRep	0.11303
TSend	0.11466
TReq	-0.05808
TWindow	0.00617
TCWLatency	0.00489
TInst	0.00256
TCheckThreshold	0.00176

$S_{mrt\ ES}(Rsp) = S_{TService}(Rsp) \times SS_{mrt\ ES}(TService)$

$S_{pCache}(Rsp) = S_{TInst}(Rsp) \times SS_{pCache}(TInst)$

Sensitivity ranking for the VM instantiation submodel

Parameter	SS(TInst)
pCache	-4.52843
vmiSize	0.52363
bw	-0.52363
t_PV	0.28465
t_CI	0.18421
t_RI	0.00752

Sensitivity ranking for the mashup sub-model

Parameter	SS(TService)
mrt_ES	0.33906
mrt_SA	0.32711
mrt_SS	0.26727
mrt_TS	0.03284
mrt_MS	0.02274
mrt_VS	0.01096

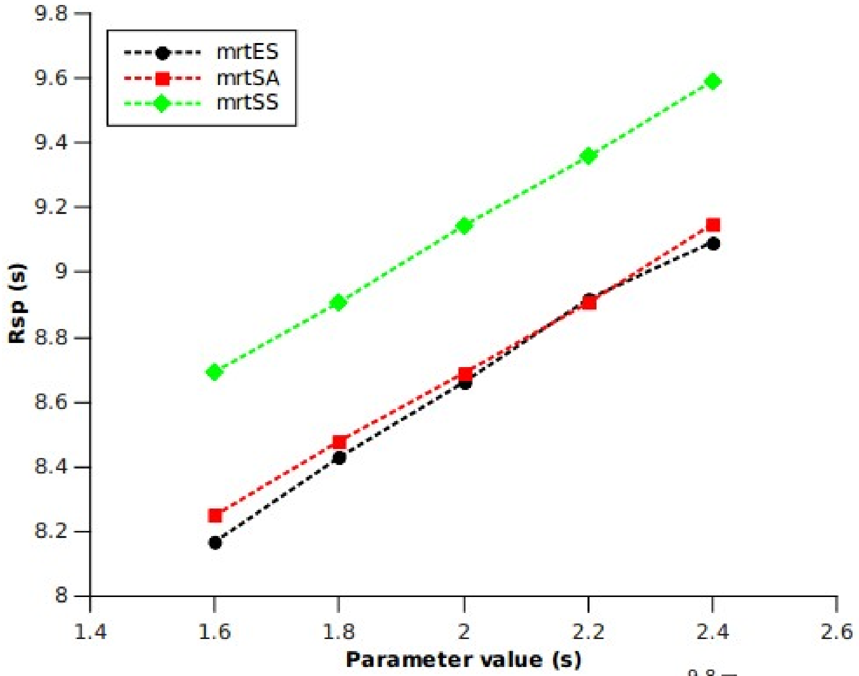
Composite web services on private cloud with autoscaling (Step 8)

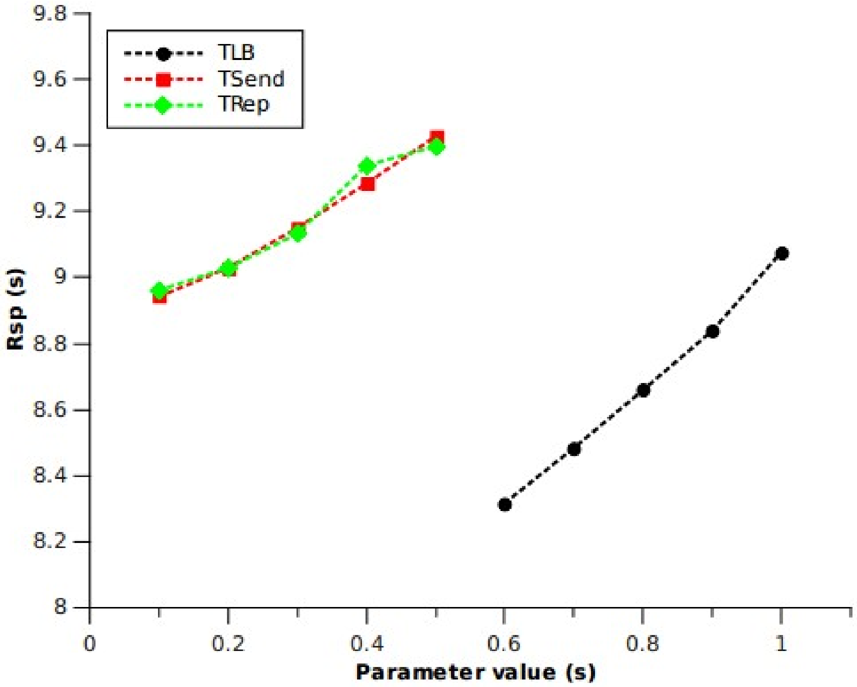
Unified sensitivity ranking for the general model and submodels

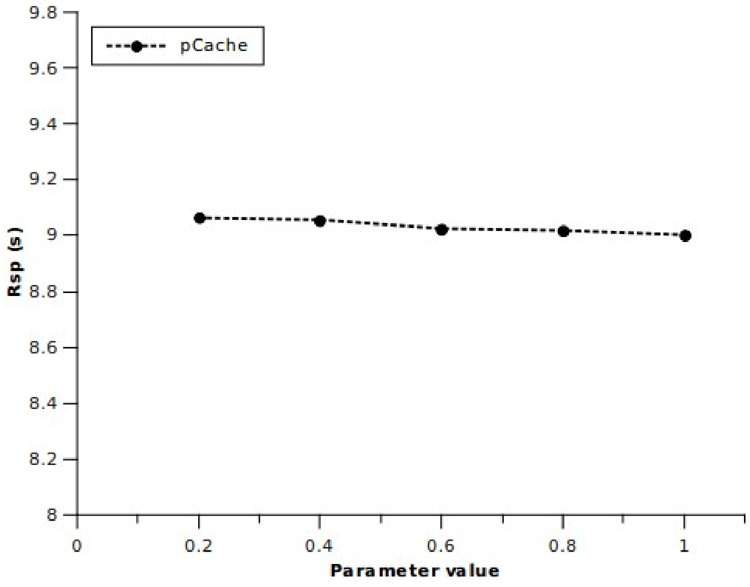
Parameter	S(Rsp)
mrt_ES	0.15517
mrt_SA	0.14969
TLB	0.13977
mrt_SS	0.12231
TSend	0.11466
TRep	0.11303
TReq	-0.05808
mrt_TS	0.01503
pCache	-0.01162
mrt_MS	0.01041
TWindow	0.00617
mrt_VS	0.00502
TCWLatency	0.00489
TCheckThreshold	0.00176
vmiSize	0.00134
bw	-0.00134
t_PV	0.00073
t_CI	0.00047
t_RI	0.00002

- Response time of following web services:
  - **Event Search**
  - **Similar Artists**
  - **Song Search**
- Execution time of **Load Balancer**
- **Network latency** for sending request and receiving reply
- The most important parameter of VM instantiation process (**pCache**) is only **intermediate** when the concern is the total response time of the application

Composite web services on private cloud with autoscaling







Universidade Federal de Pernambuco

Final remarks: Achieved results

- Methods for building **unified sensitivity rankings**, considering composition of RBD with CTMC sub-models, and also SPN with CTMC sub-models.

- **Supporting methodology** for identification of performance and availability bottlenecks in cloud computing systems. Efficacy demonstrated in case studies.

- Hierarchical **availability models** that enable evaluating **private cloud** and *mobile cloud* architectures.

Final remarks: Achieved results

- Hierarchical **performance models** that enable planning **composite web services** hosted on clouds with **elasticity** and load balancing mechanisms.

- **Automated sensitivity analysis** features for hierarchical models in Mercury tool

- **Sensitive GRASP** algorithm, for optimizing performance and availability metrics of cloud-hosted services.