contrail

open computing infrastructures for elastic services

ConPaaS: A Cloud Platform for Hosting Elastic Applications

Héctor Fernández and G. Pierre
Vrije Universiteit Amsterdam

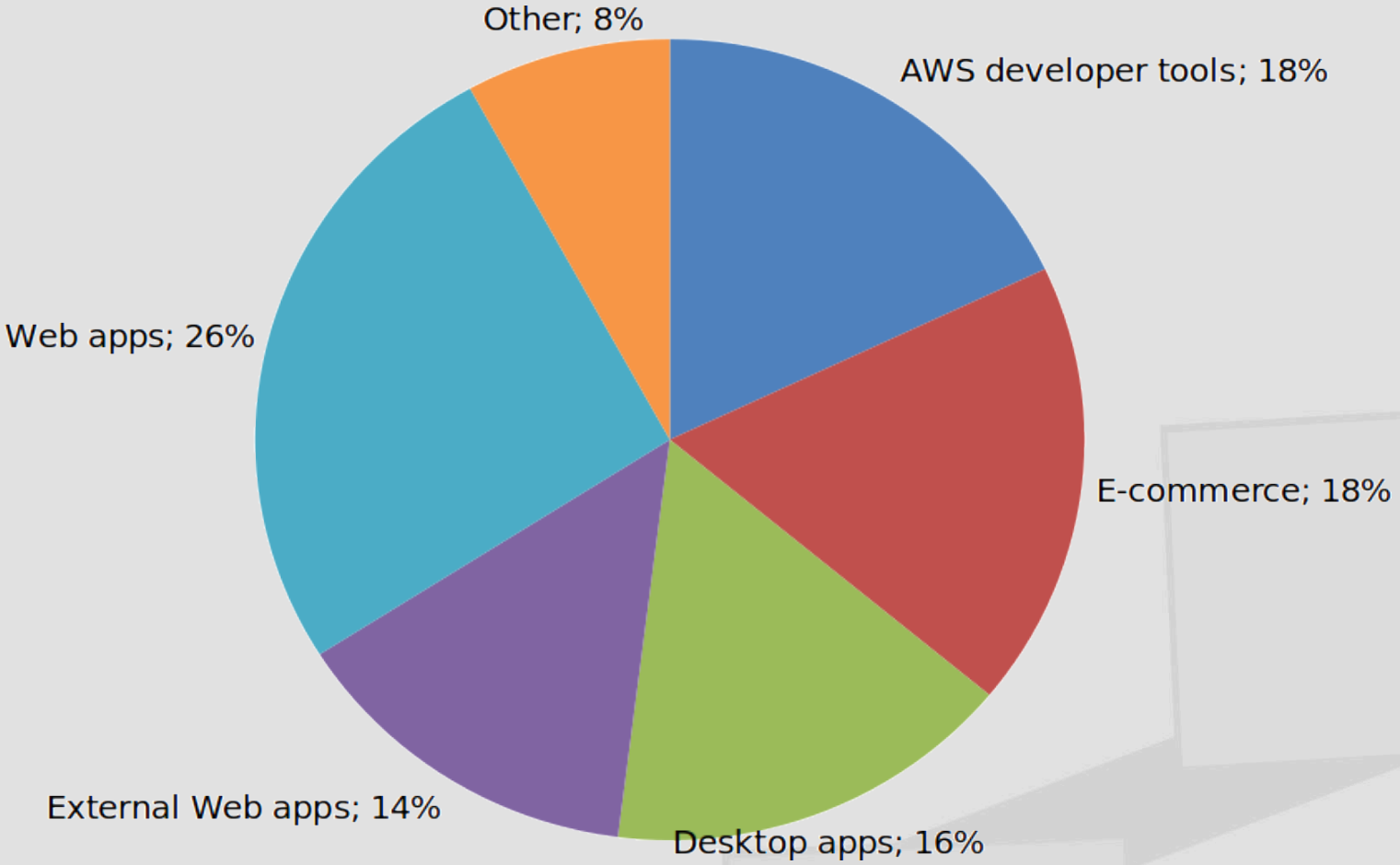
Cloud Computing Day, November 20th 2012

contrail is co-funded by the EC
7th Framework Programme
under Grant Agreement nr.
257438

Typical Cloud Applications (according to AWS)

- Application Hosting
- Backup and Storage
- Content Delivery
- E-Commerce
- High Performance Computing
- Media Hosting
- On-Demand Workforce
- Search Engines
- Web Hosting

Applications running at Amazon Web Services



Sample: 50 applications from the AWS Customer App Catalog.

Many Cloud applications are alike

- Web servers
- Application servers
- Database servers
- High-performance frameworks (MapReduce, MPI, Workflows)
- . . . and a few percents of miscellaneous programs

Cloud application developers often rebuild
the same types of frameworks again and again and again. . .

Can the Cloud help support common types of applications?

- Infrastructure-as-a-Service provides basic computing resources
  - Absolute flexibility: you can build anything you want
  - But it can be very complex and time consuming
    - Deployment
    - Software upgrades
    - Fault-tolerance
    - Performance monitoring
    - Resource provisioning
    - Dynamic reconfiguration orchestration
    - etc.

- Platform-as-a-Service provides high-level services
  - Each PaaS service targets a specific family of applications
  - Provide a simple deployment environment for applications
  - Provide high-level guarantees for applications using these services

What is ConPaaS ?

ConPaaS is an open-source runtime environment for hosting applications in Cloud environments.

- Broad range of functionalities
  - Application servers, databases, high-performance computing, miscellaneous
- Fully integrated
  - Applications can compose any set of services together
- Easy to use but also very powerful
  - Simple Web GUI + powerful command-line tool
  - Services are highly customizable
- Cutting-edge SLA enforcement technologies
  - Elasticity and resource provisioning techniques to guarantee performance at the lowest possible cost
- Support for deployment on multiple-clouds
  - OpenNebula, AmazonEC2, OpenStack (soon)

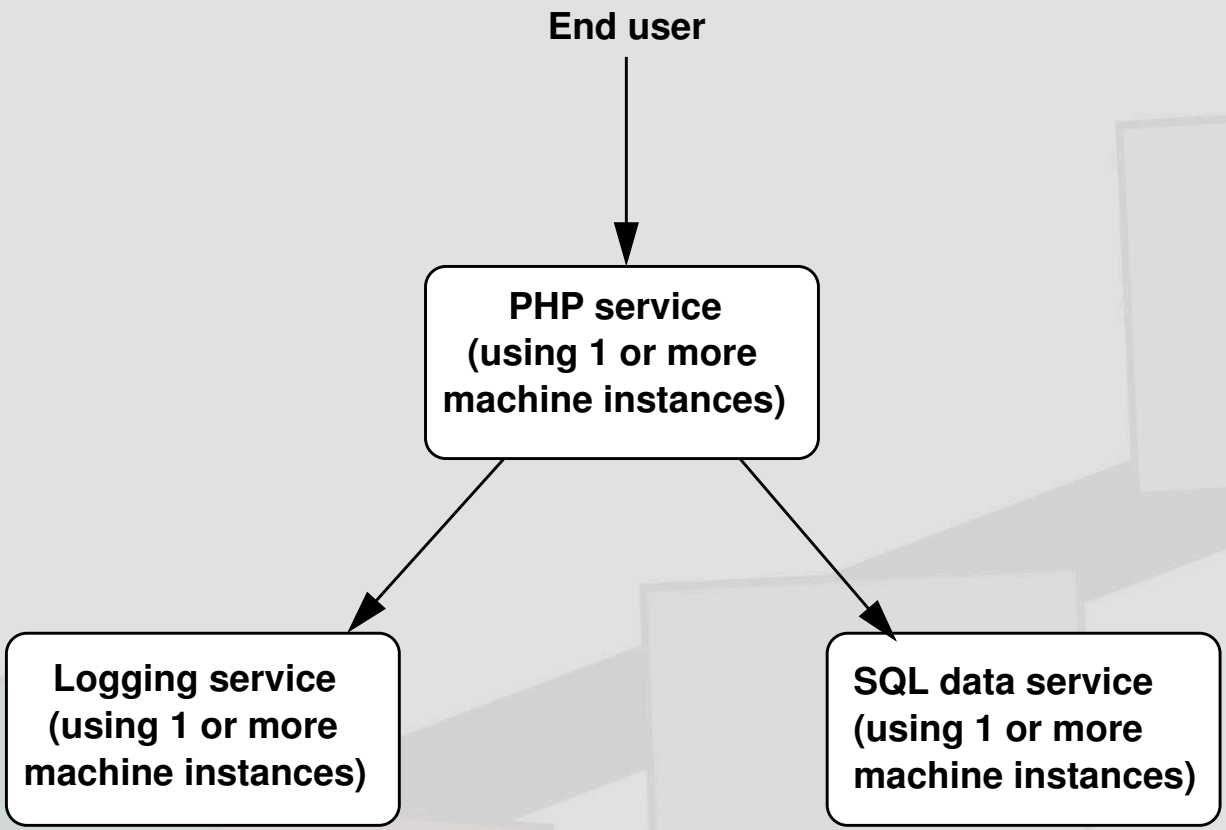
ConPaaS hosting Cloud Applications

- Fully support for the following services:
  - Web servers — static content and dynamic web applications (PhP, JSP)
  - MapReduce — for data-intensive computing
  - TaskFarming — for scientific applications
  - Databases (SQL and NoSQL) — for everybody
  - More services coming: CDN, functional testing, XtreemFS, etc.
- **BUT:** You can easily build your own ConPaaS service.

ConPaaS Applications

A ConPaaS application is defined as a composition of multiple service instances

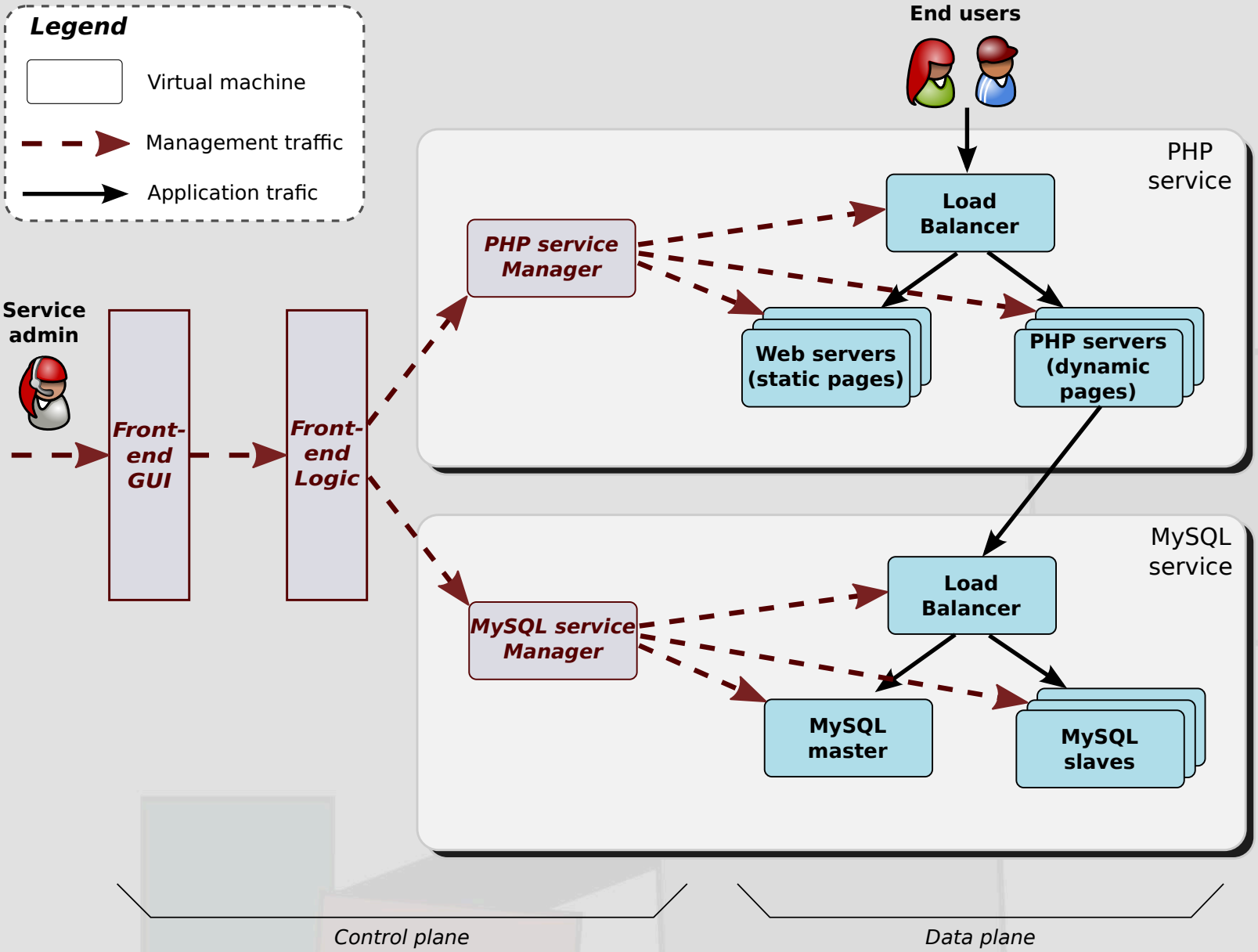
- For example: web hosting service + MySQL database + logging service (to store access logs)



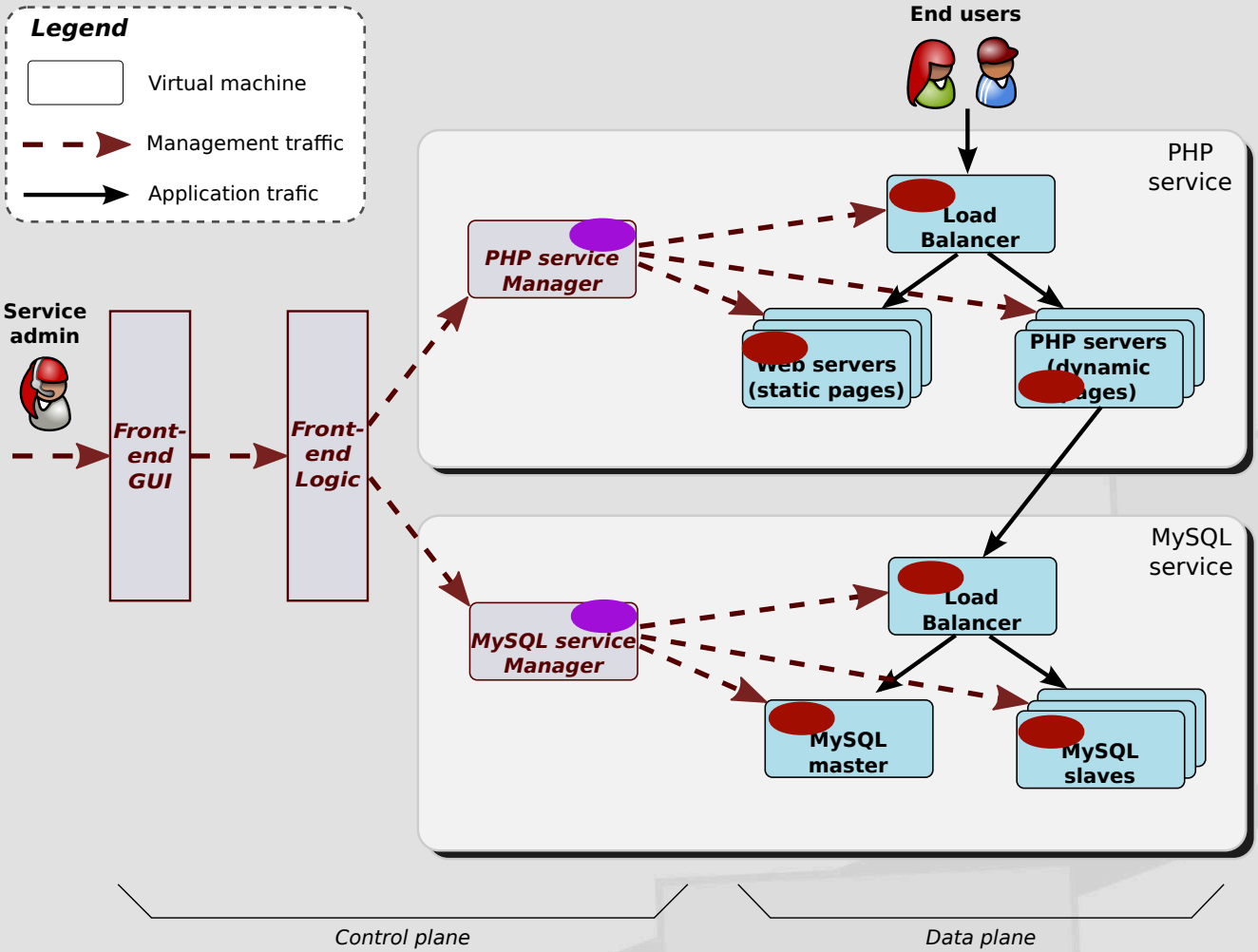
Architecture of a ConPaaS service

- A ConPaaS service is implemented as one or more virtual machine instances dedicated to a single user
  - Single-tenant: each VM belongs to a single user
  - No VM sharing between services (even for the same user)
- ConPaaS services are elastic: we can grow/shrink their capacity at runtime with no service disruption
  - Horizontal provisioning: add/remove virtual machines
- ConPaaS services will support dynamic resource provisioning: automatic capacity adjustment to support performance guarantees at minimum cost

ConPaaS Organization

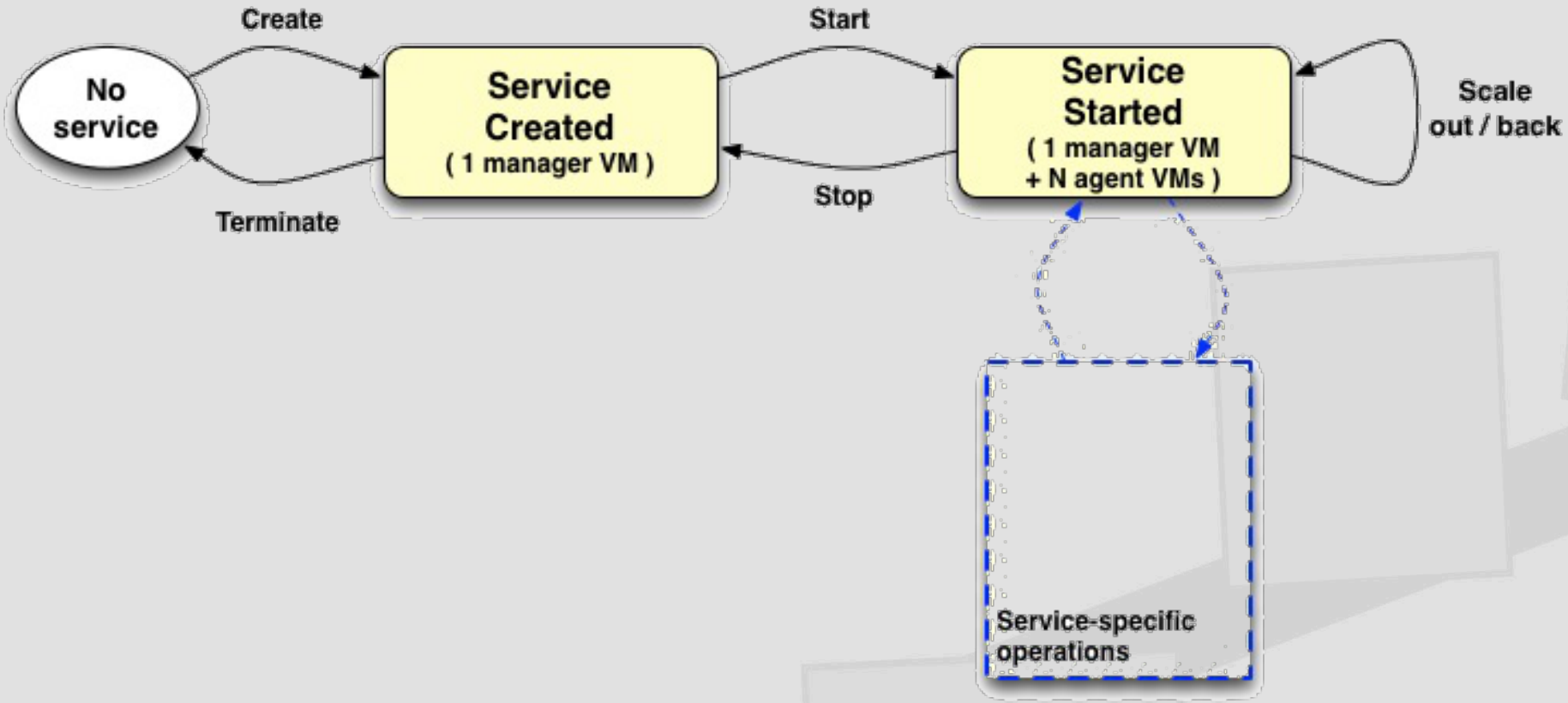


ConPaaS Organization

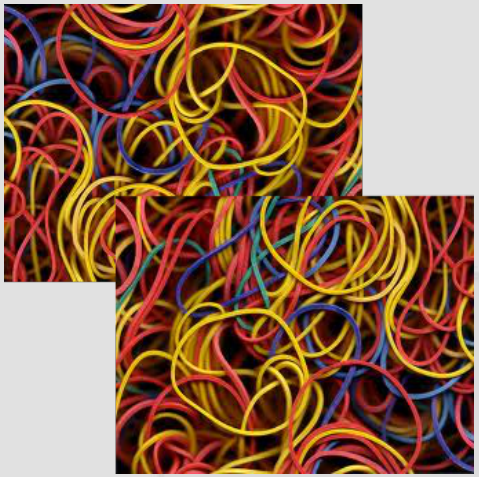


- Each manager VM contains a manager process
- Each agent VM contains an agent process

Lifecycle of a ConPaaS service

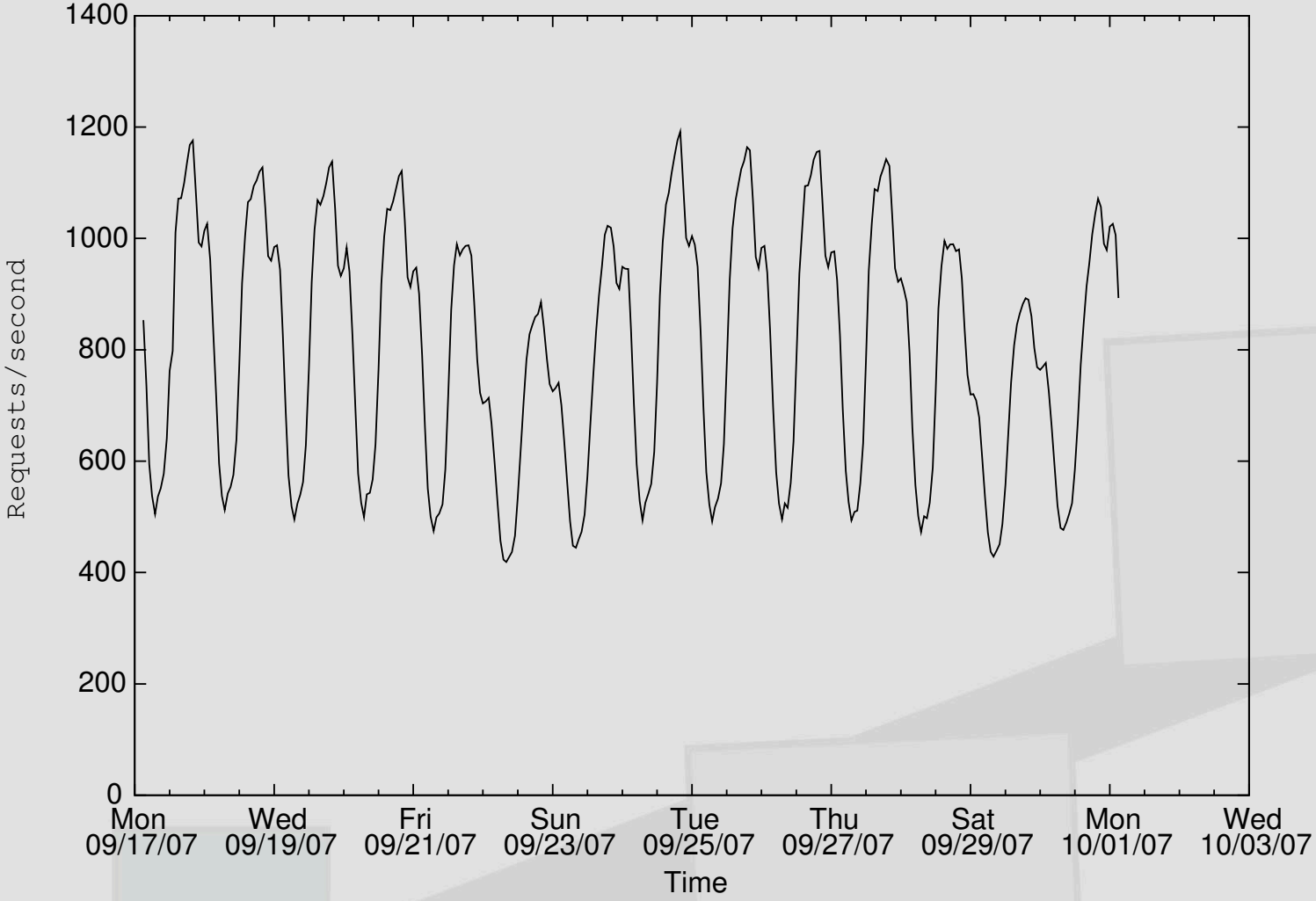


Elastic Cloud Applications

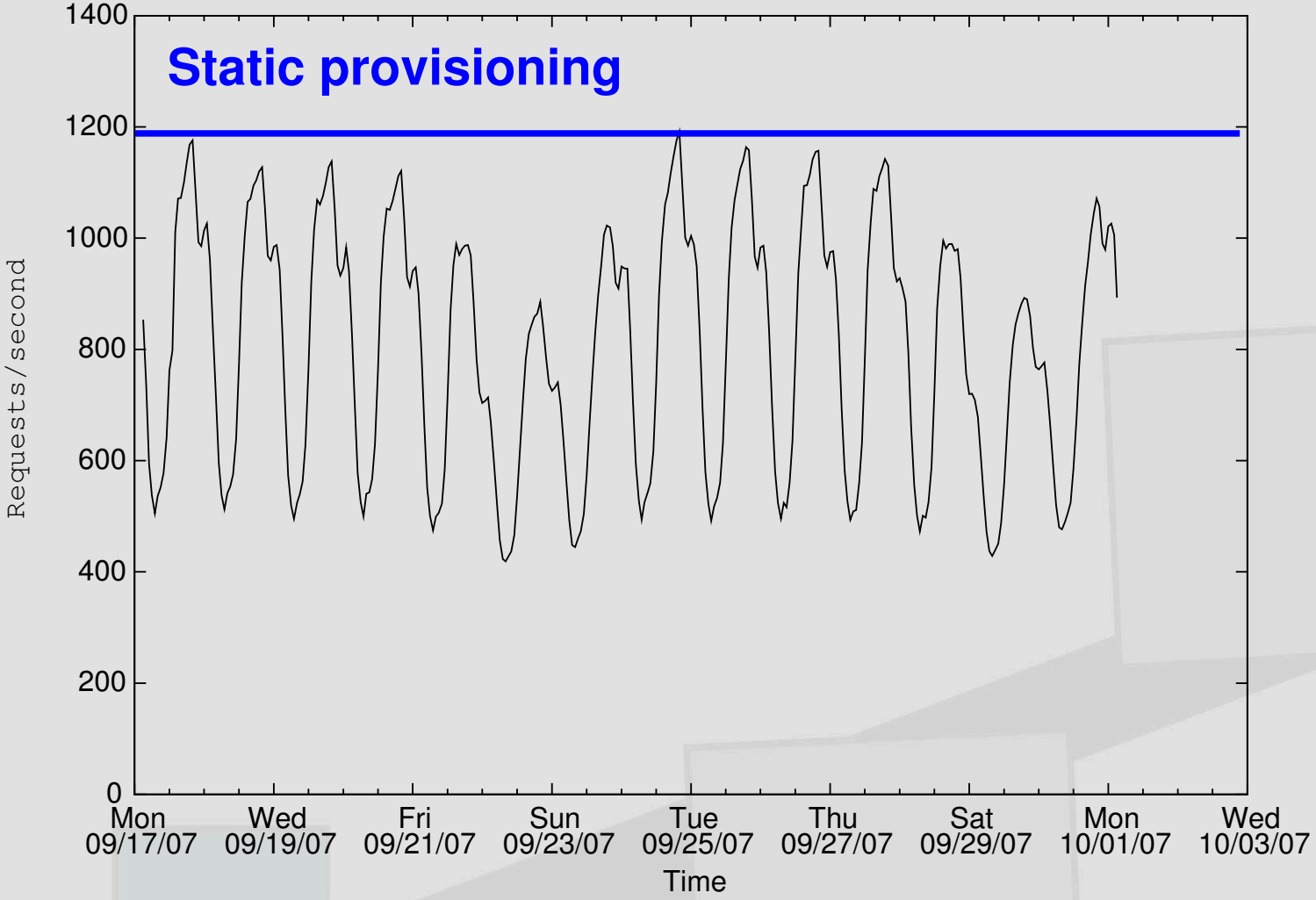


– Web applications

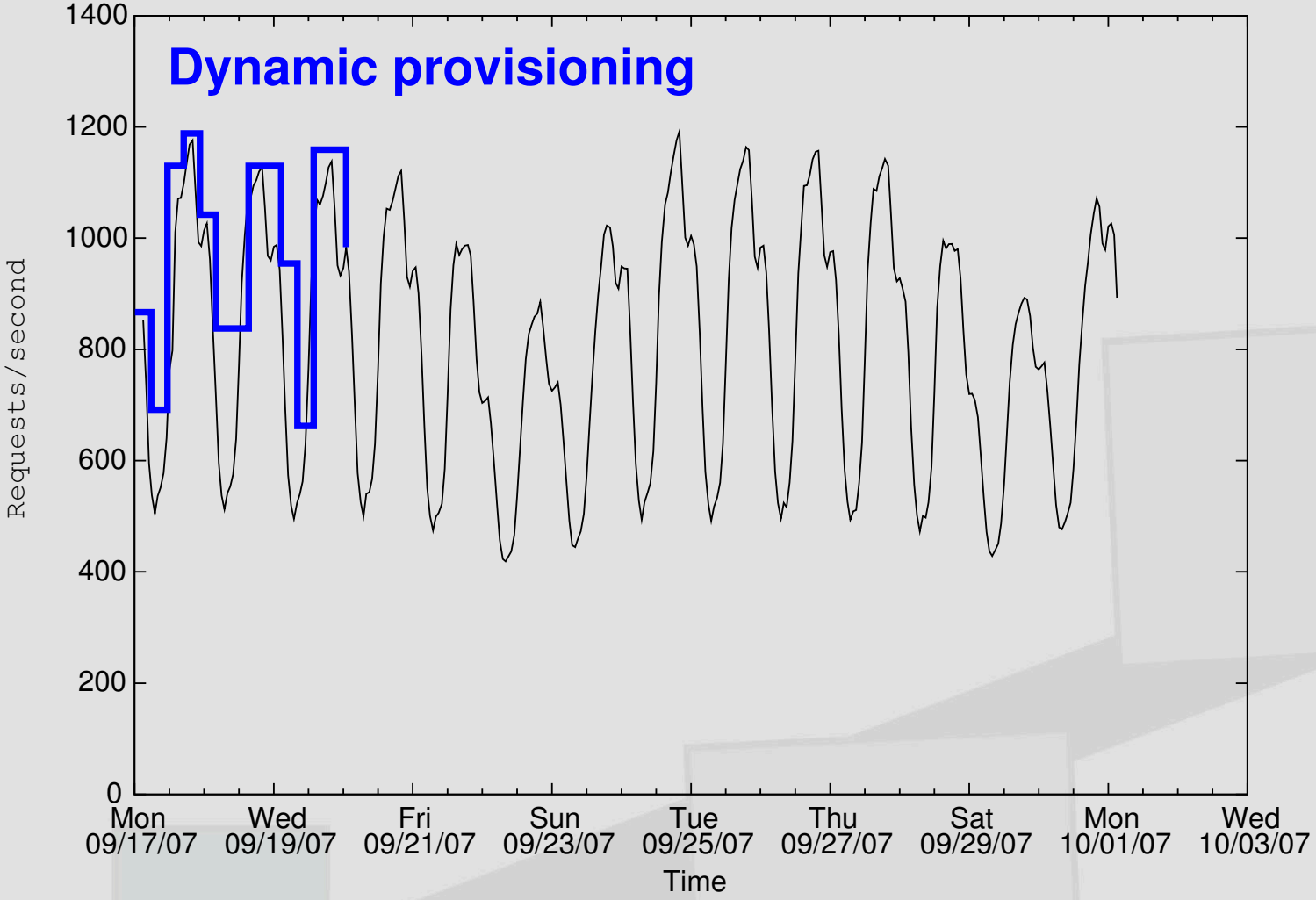
The varying capacity problem



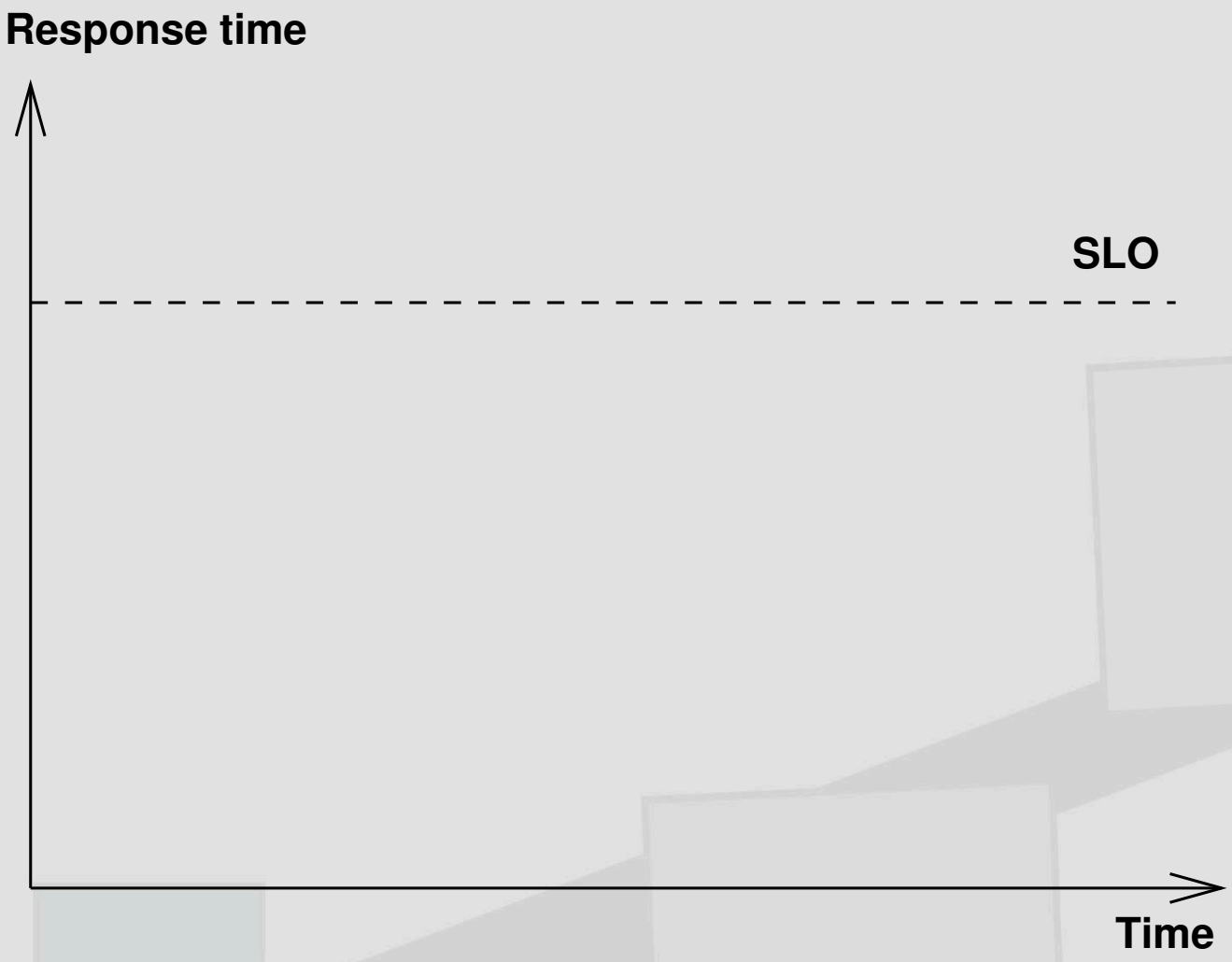
The varying capacity problem



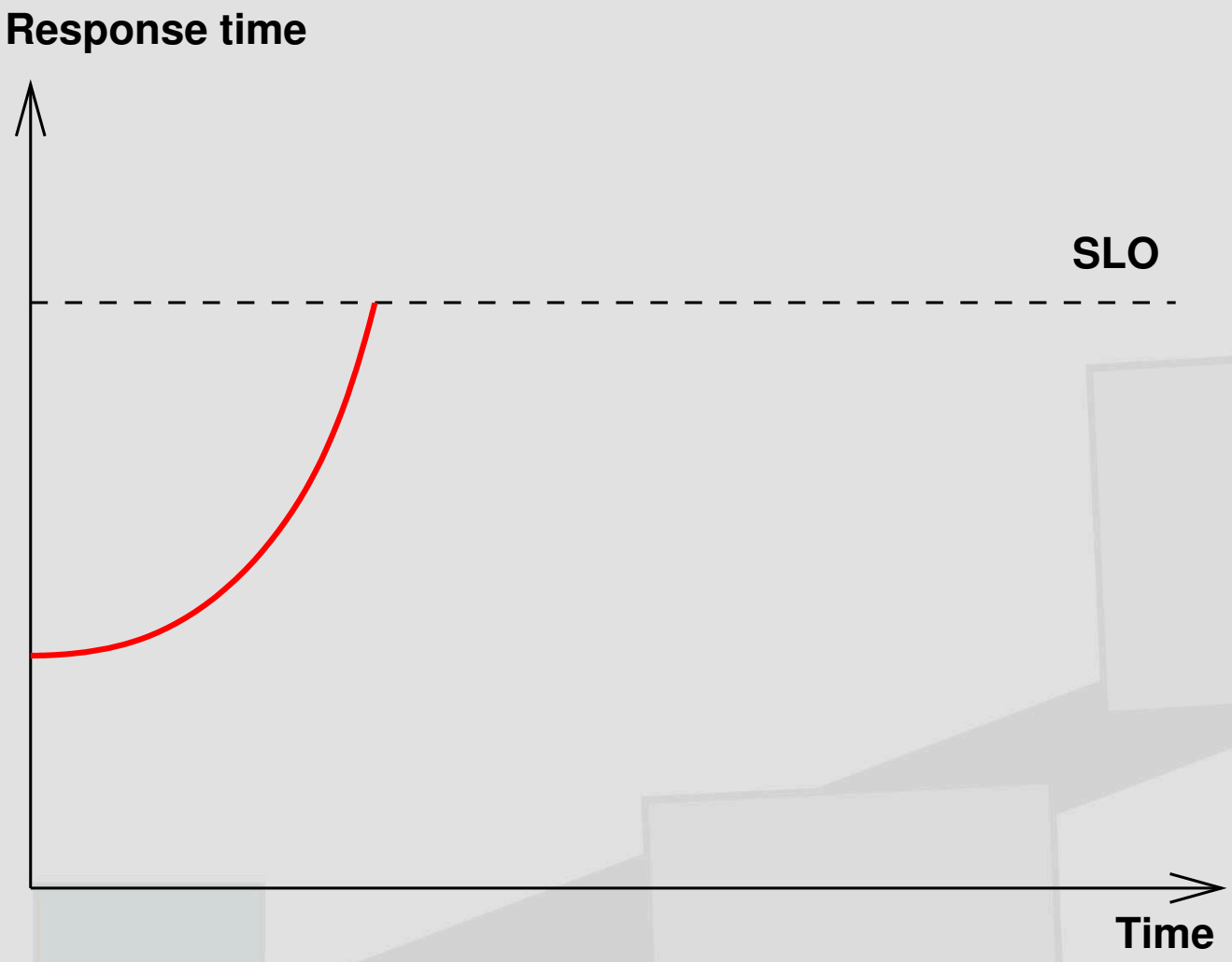
The varying capacity problem



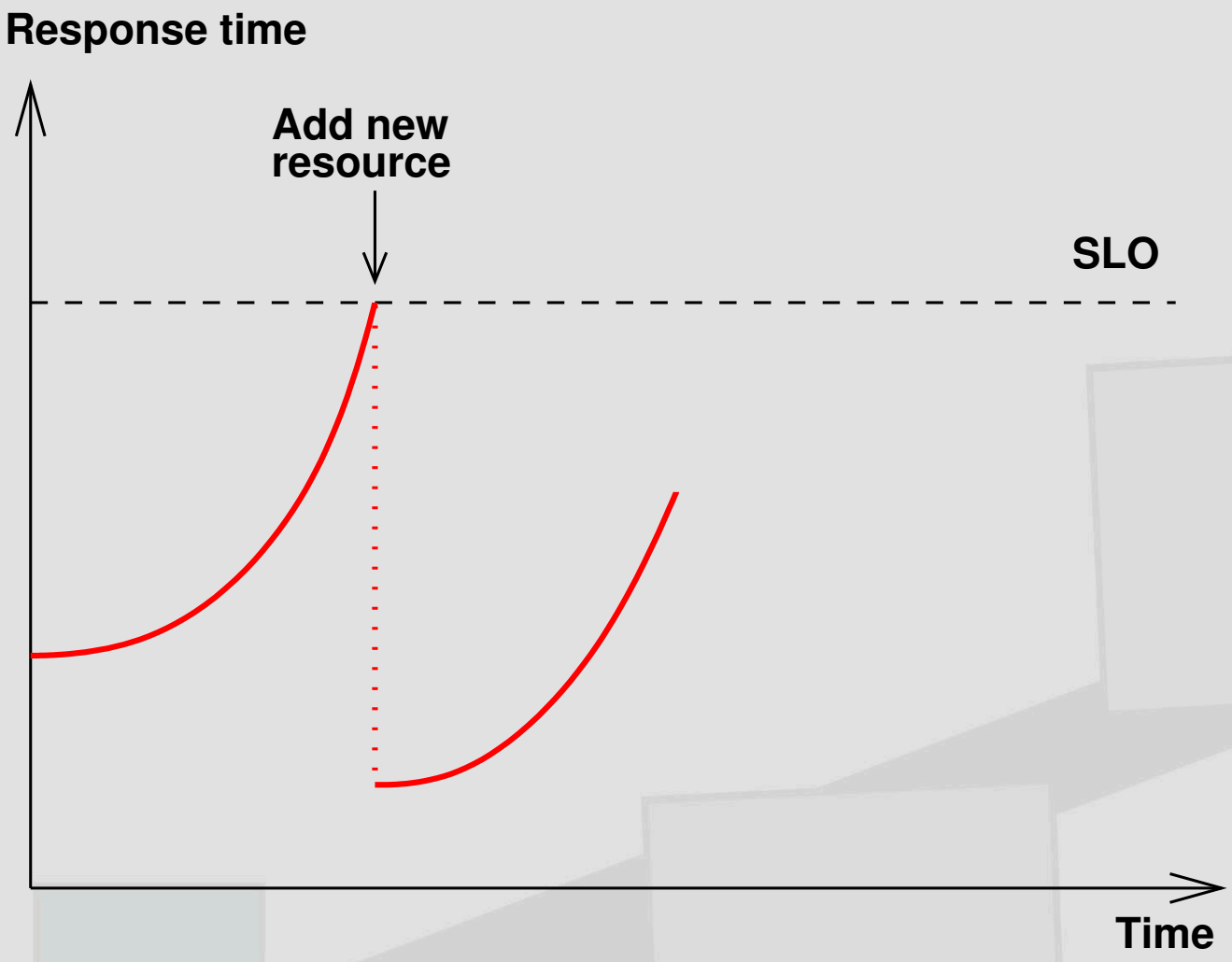
Dynamic resource provisioning



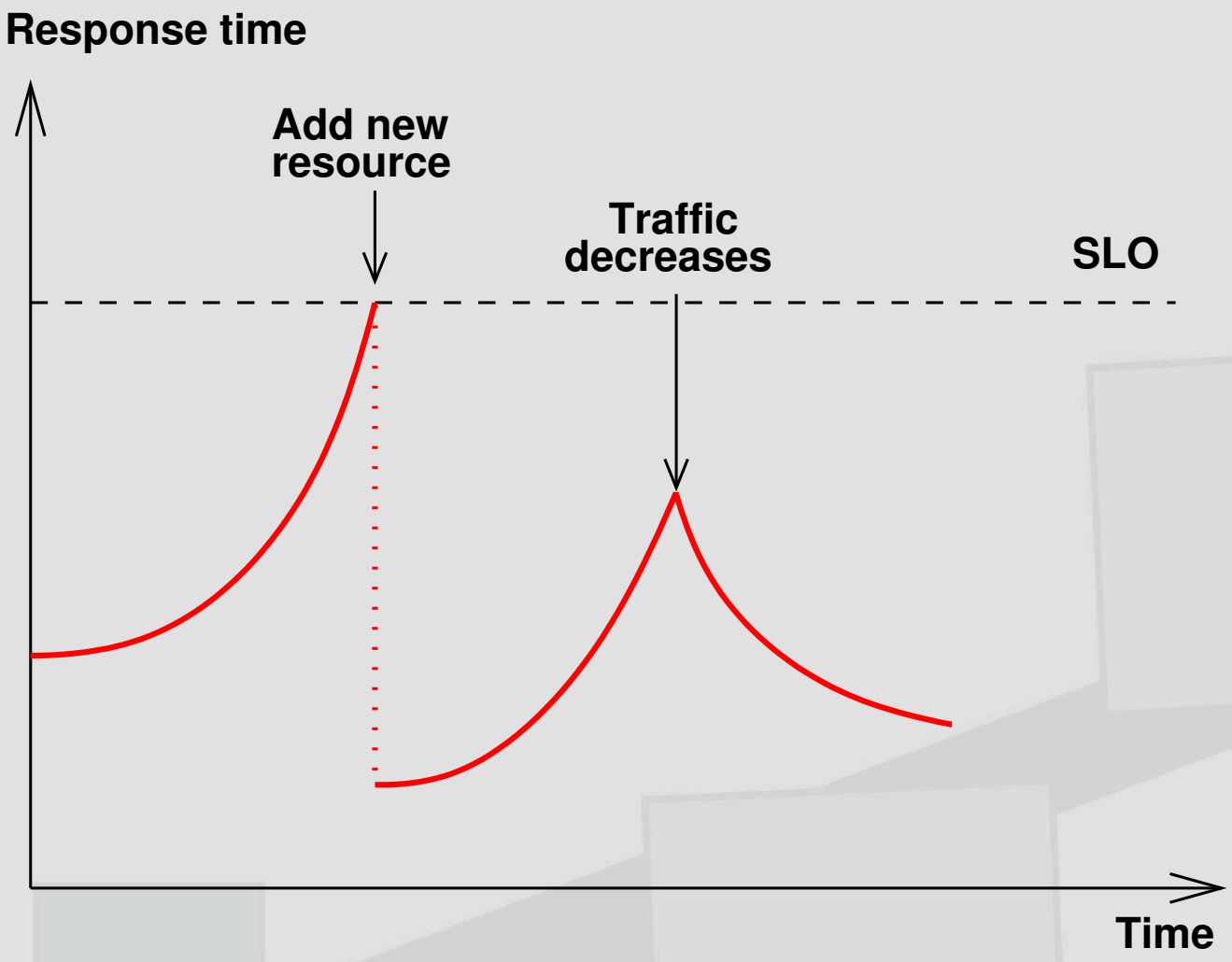
Dynamic resource provisioning



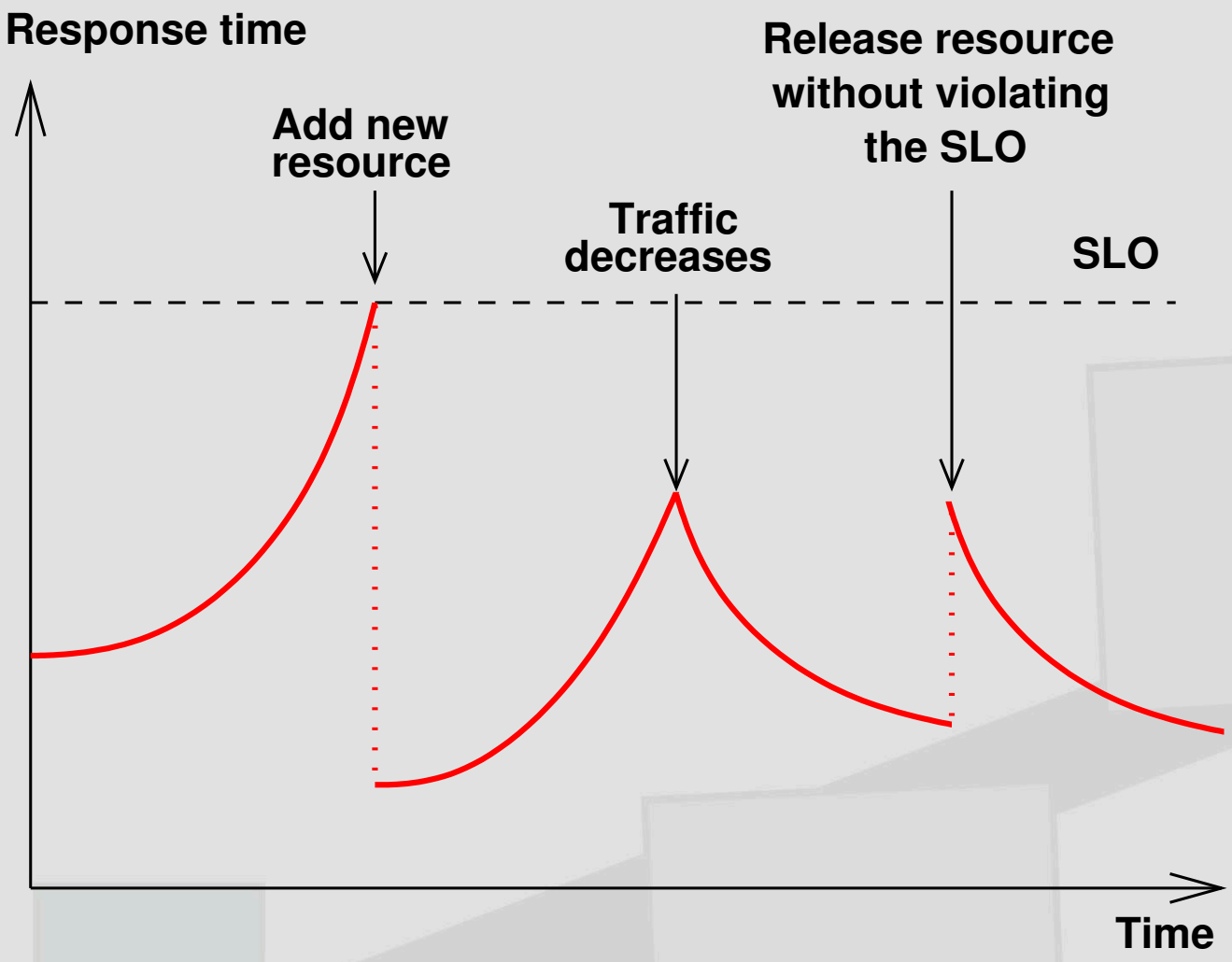
Dynamic resource provisioning



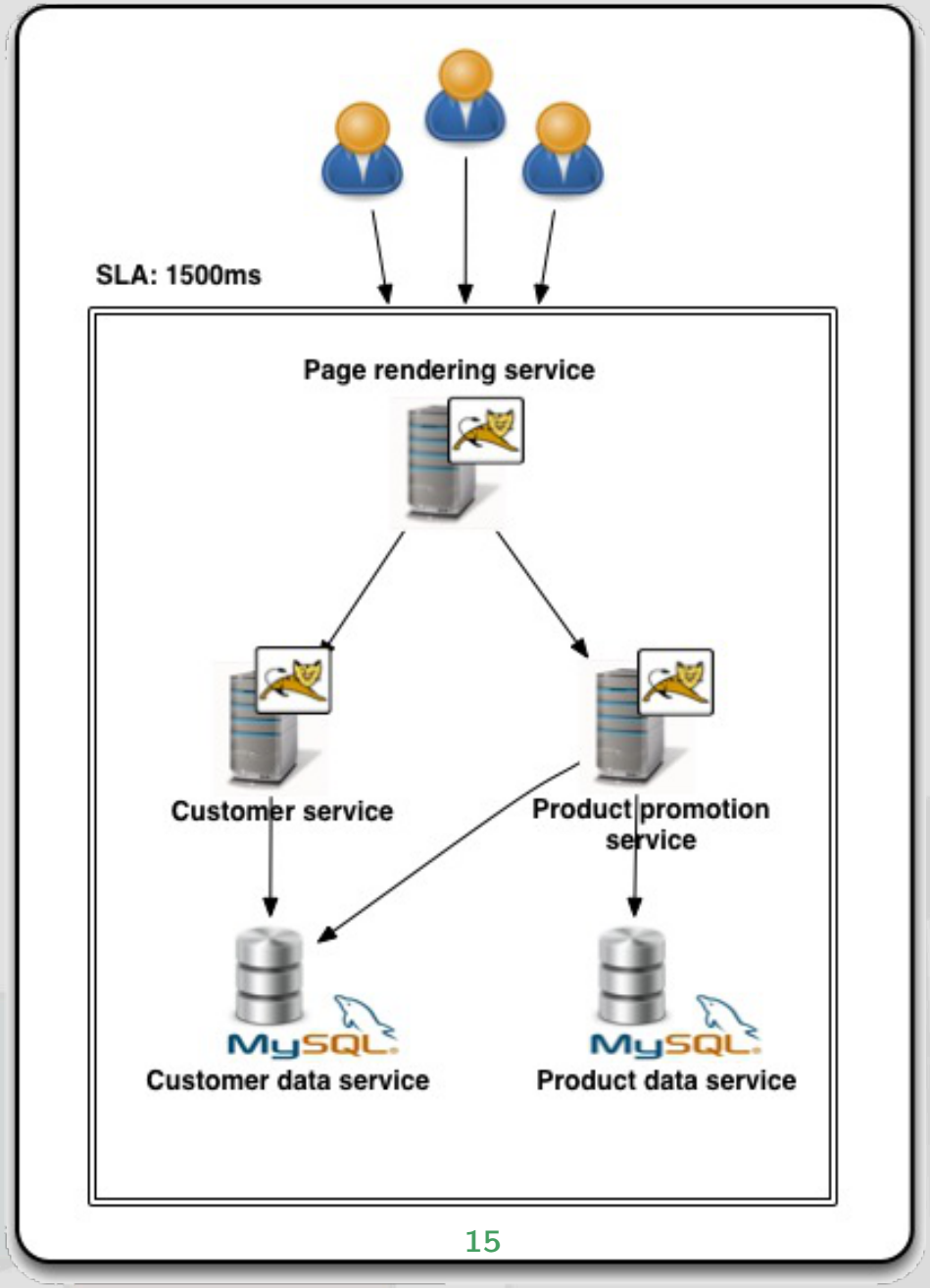
Dynamic resource provisioning



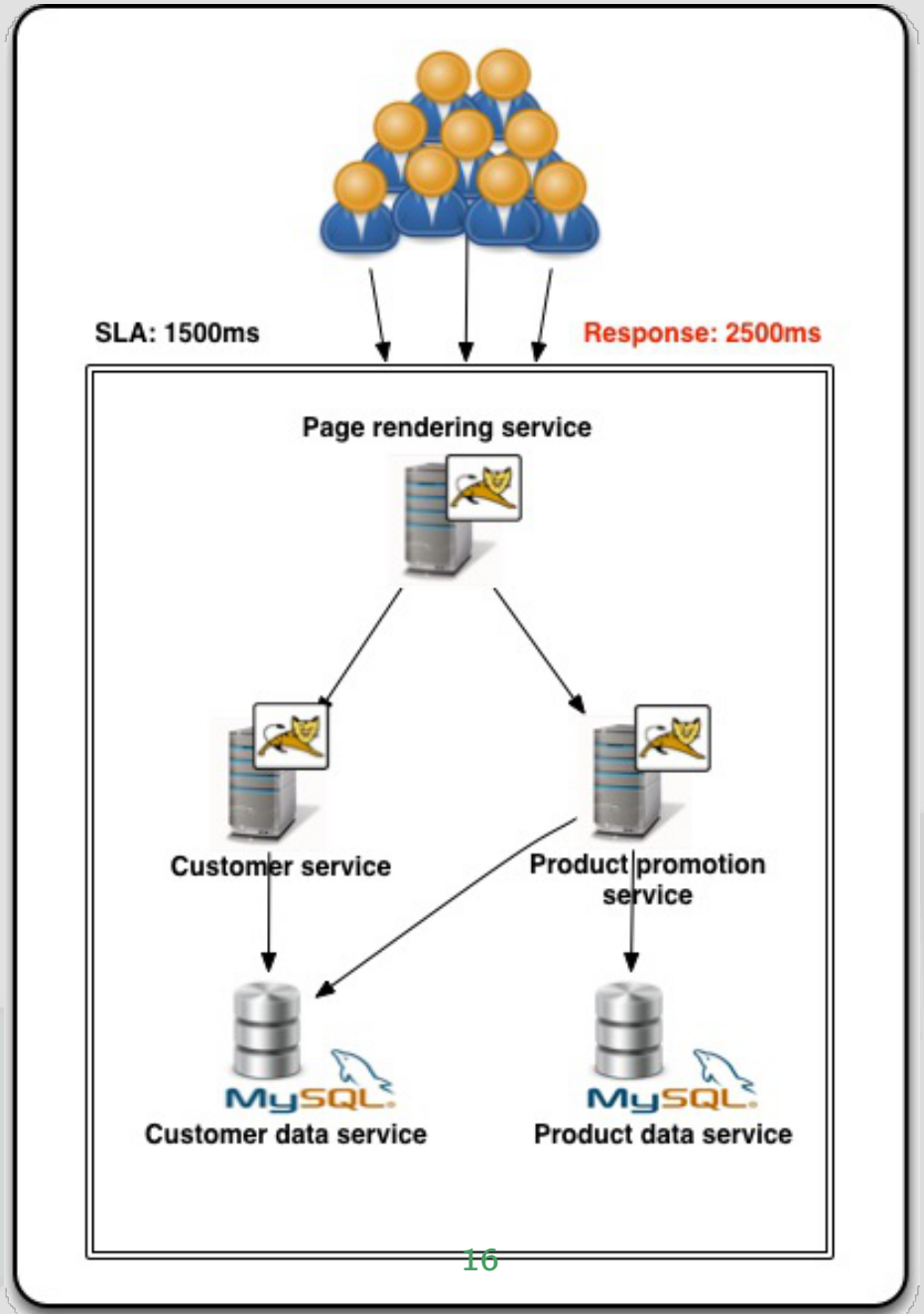
Dynamic resource provisioning



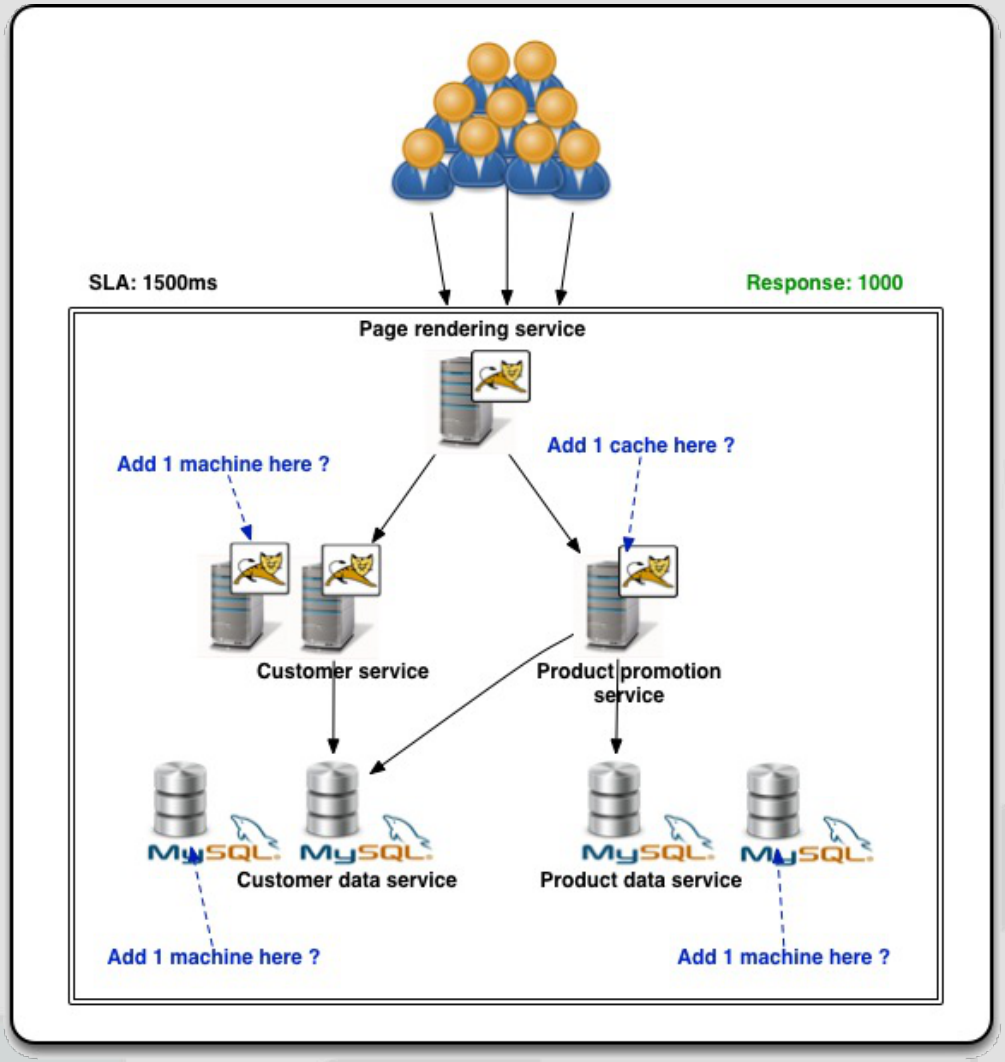
Provisioning in a multi-service Web application



Provisioning in a multi-service Web application



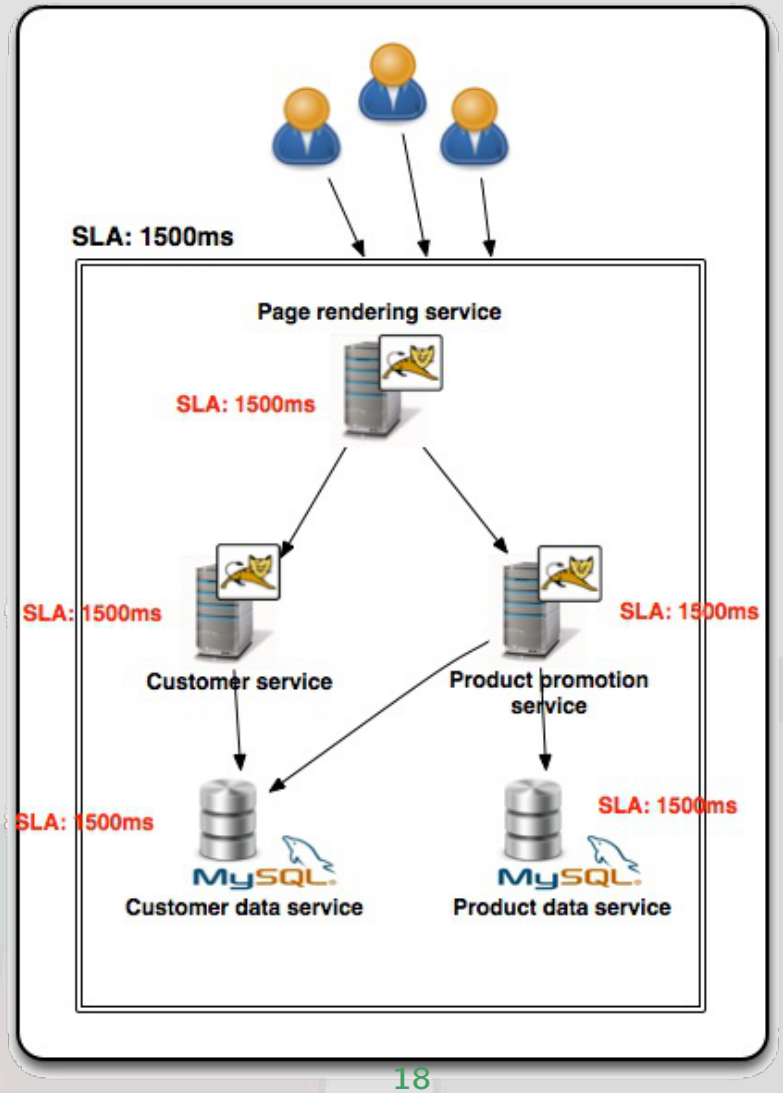
Provisioning in a multi-service Web application



When the application needs more capacity, where should I place extra resources?

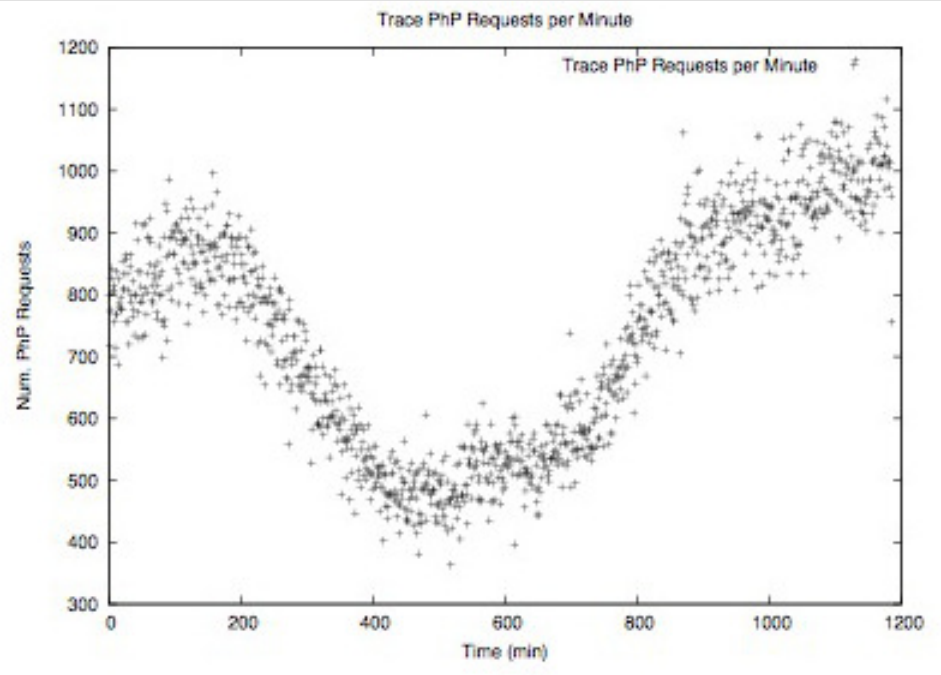
Classical solutions

- Threshold rules-based resource provisioning systems
  - Impose an SLO to each service individually



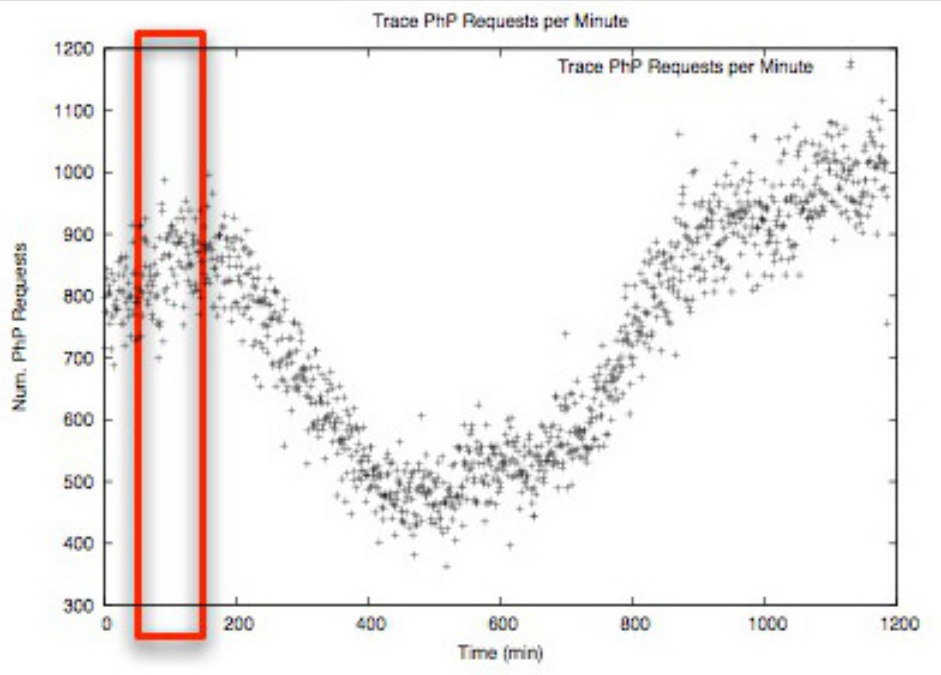
Classical solutions

- Threshold rules-based resource provisioning systems
  - Impose an SLO to each service individually
  - Scaling decisions are based on the actual workload burst



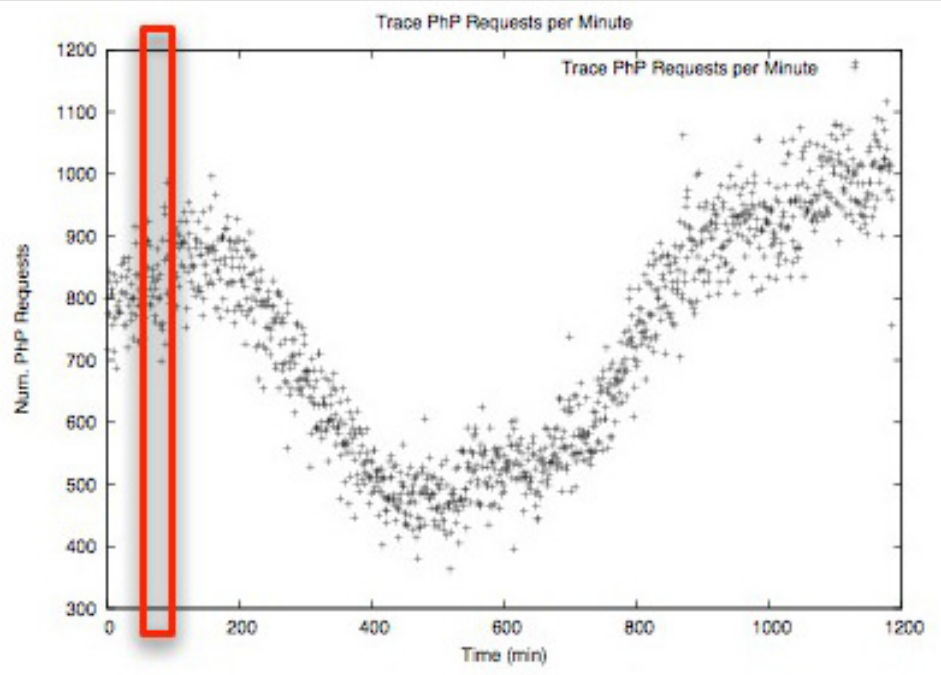
Classical solutions

- Threshold rules-based resource provisioning systems
  - Impose an SLO to each service individually
  - Scaling decisions are based on the actual workload burst



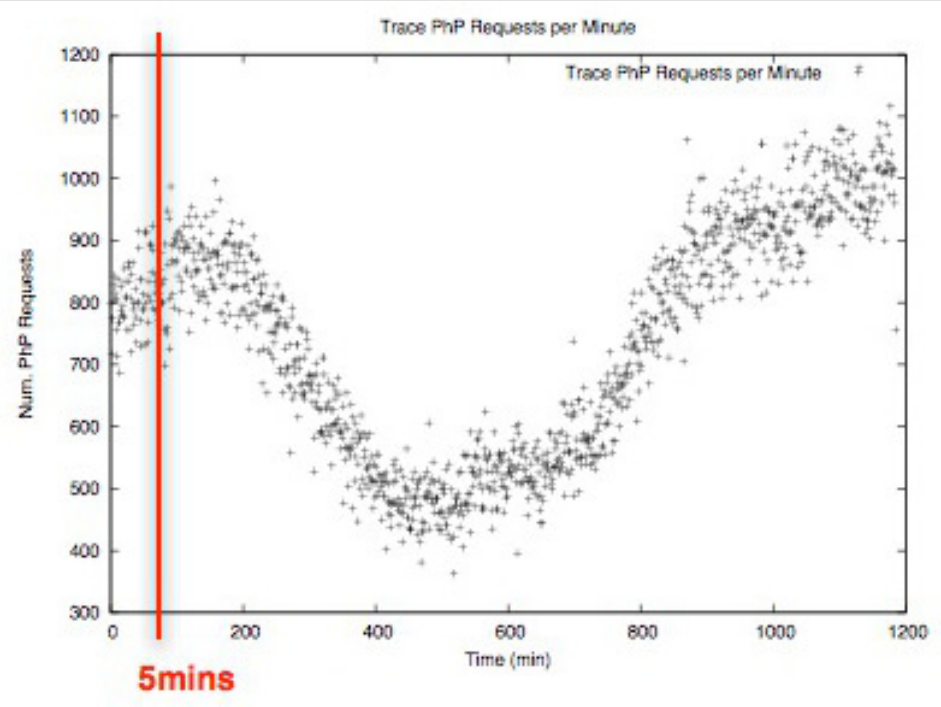
Classical solutions

- Threshold rules-based resource provisioning systems
  - Impose an SLO to each service individually
  - Scaling decisions are based on the actual workload burst



Classical solutions

- Threshold rules-based resource provisioning systems
  - Impose an SLO to each service individually
  - Scaling decisions are based on the actual workload burst



Classical solutions

- Threshold rules-based resource provisioning systems
  - Impose an SLO to each service individually
  - Scaling decisions are based on the actual workload burst
  - Examples: RightScale, OpenShift, Auto Scaling Amazon, etc ...

- Advantages
  - Each service can be scaled in isolation
  - Easily interpreted by non specialists
- Disadvantages
  - Only system-level performance constraints (CPU usage and Resp. time)
  - Easy target to temporal bursty variations in the workload
  - VM performance heterogeneity
  - Overall performance is suboptimal: SLO for backend services

ConPaaS: Auto-scaling System

- Profiling-based resource provisioning system
  - Extend the threshold rules-based system using profiling techniques
    - VM performance profiling
  - Impose an SLO to the front-end service only
  - Scaling decisions are based on the most recent workload history
- Advantages
  - Services collaborate to maintain the SLO at min. cost
  - VM performance heterogeneity
  - Application-specific performance constraints
  - Avoid flash crowds and slashdot effects
- Disadvantages
  - Best-effort in terms of SLA fulfillment

VM performance profiling

Purpose: To estimate the application response time that a certain VM will provide under a given workload.

Online profiling:

- Idea: while the application is in use, we direct a number of specified request workloads to the tested VMs and measure the response time
- Usage:
  - to dynamically adjust the load balancing weights of the provisioned VM's
  - to refine the conditions for scaling out/back
- Implementation: through a customized web load balancer

VM performance profiling

Purpose: To estimate the application response time that a certain VM will provide under a given workload.

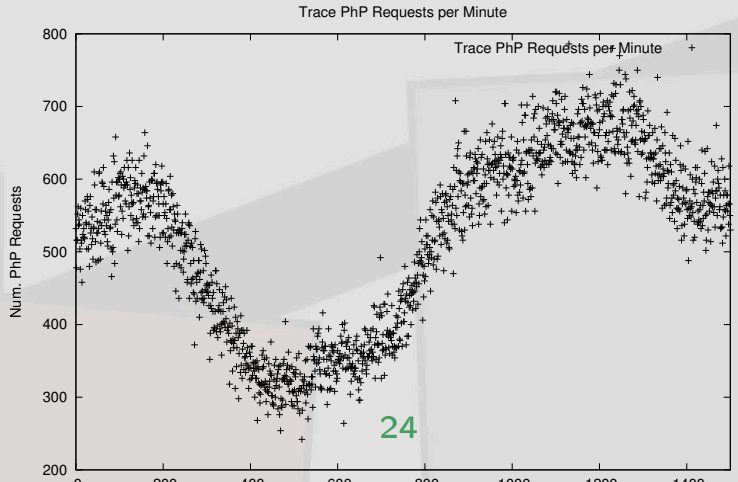
Offline profiling:

- Idea: to gather information about VM instances to have an initial assessment of their throughput.
- Usage:
  - to select the suitable set of VMs for an initial configuration
  - to define the flexible threshold ranges for each VM's instance
- Implementation: through training

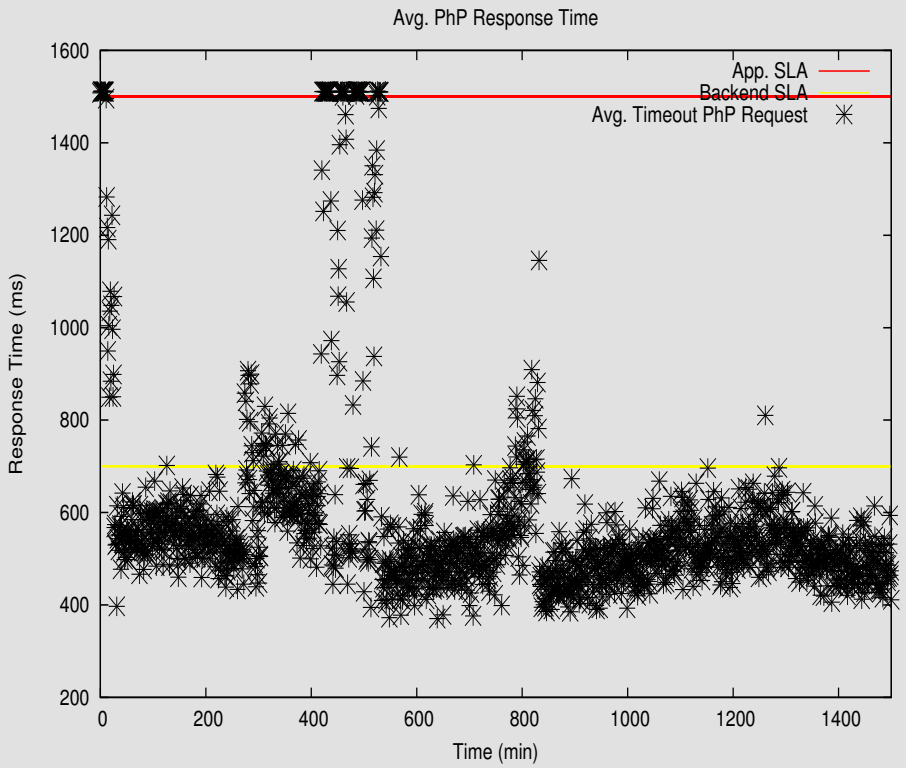
Preliminary Experimental Results

Purpose: Compare the behavioral pattern between the threshold-based and profiling-based (using the PhP web hosting service)

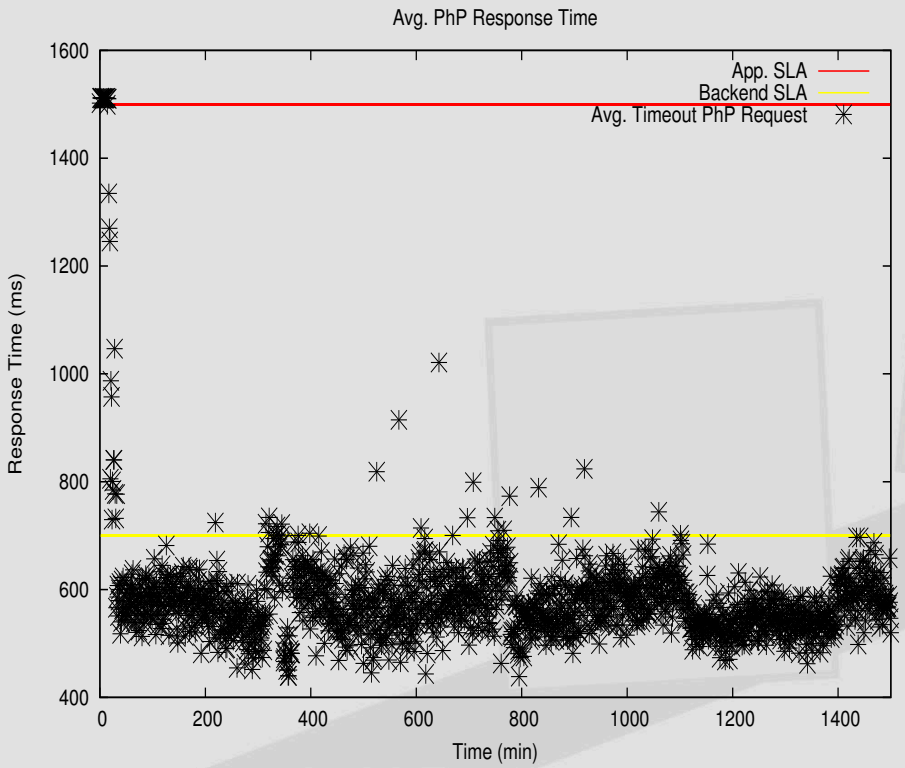
- Application: Wikipedia services - Mediawiki
  - English Wikipedia articles data (approx. 20Gb)
- Monitoring:
  - Ganglia (http://ganglia.sourceforge.net/)
  - Modules to monitor web-specific metrics
- Testing: Wikibench, a Wikipedia-based benchmark
  - Real access traces
- Enviroment: Amazon Elastic Cloud Compute



SLA Fulfillment (only PhP requests)

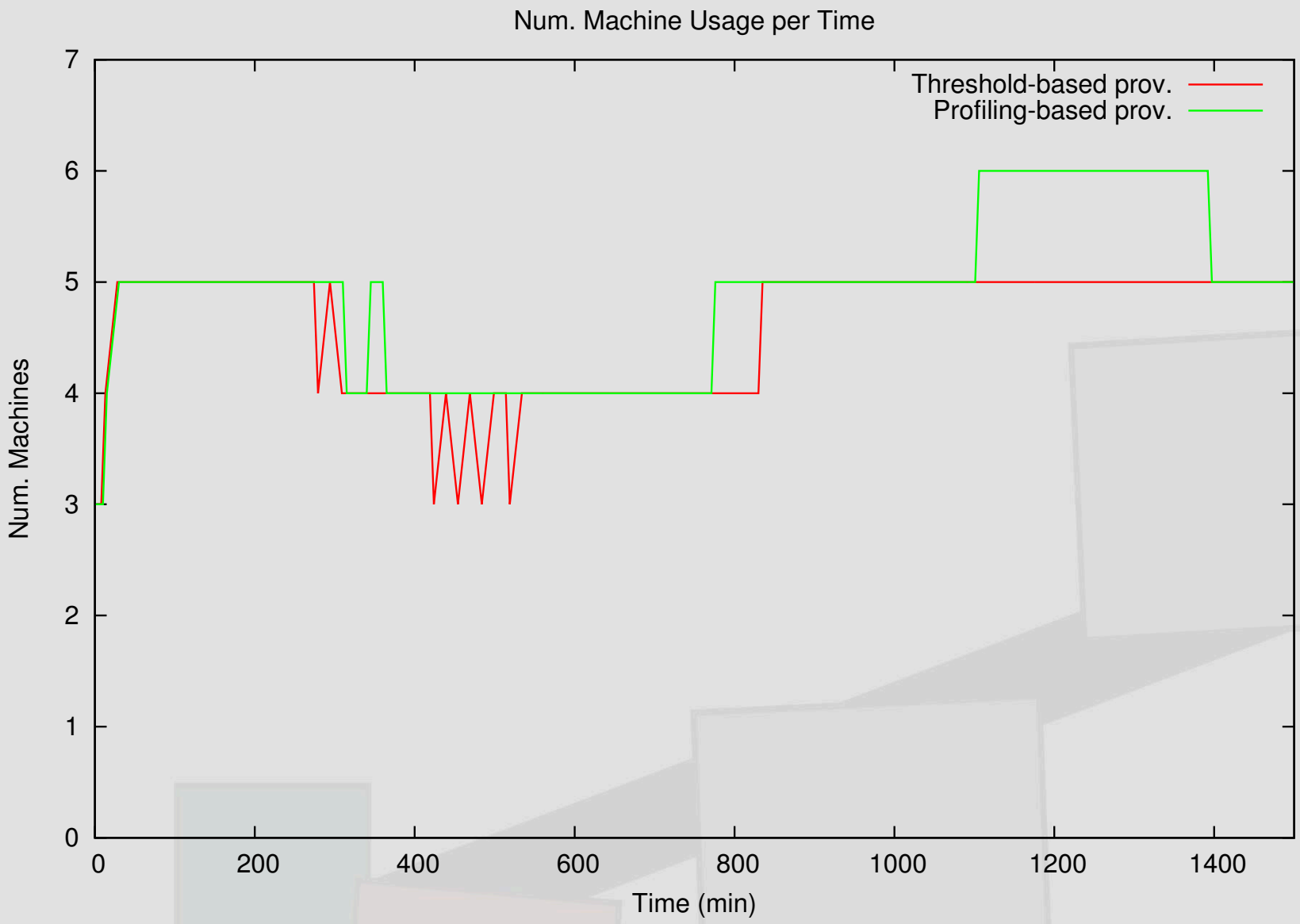


Threshold rules-based



Profiling-based

Resource Consumption



Conclusion

- ConPaaS is a platform-as-a-service environment
  - Designed to facilitate elastic application hosting in the cloud
    - Auto scaling system: Trade-off between SLA fulfillment and resource consumption
  - Designed to be easily extensible

- ConPaaS addresses two major classes of applications:
  - Web applications
  - Scientific applications
  - Combinations of both

- **ONGOING WORK**:
  - Online profiling when adding/removing VMs
  - Offline profiling to establish flexible threshold ranges
  - Vertical scaling (up/down)
  - Cost-aware resource provisioning

contrail

open computing infrastructures for elastic services

contrail is co-funded by the
EC 7th Framework Programme

Funded under: FP7 (Seventh Framework Programme)
Area: Internet of Services, Software & Virtualization (ICT-2009.1.2)
Project reference: FP7-IST-257438
Total cost: 11.29 million euro
EU contribution: 8.3 million euro
Execution: From 2010-10-01 till 2013-09-30
Duration: 36 months
Contract type: Collaborative project (generic)

The ConPaaS Front-End

ConPaaS - management in

10.100.0.97:9999/login.php

ConPaaS

ConPaaS is the Platform-as-a-Service component of the Contrail E.U. project.

ConPaaS aims at facilitating the deployment of applications in the cloud. It provides a number of services to address common developer needs. Each service is self-managed and elastic:

- it can deploy itself on the cloud
- it monitors its own performance
- it can increase or decrease its processing capacity by dynamically (de-)provisioning instances of itself in the cloud

Login

username

password

login register

The ConPaaS Front-End

ConPaaS - management in

10.100.0.97:9999/login.php

ConPaaS

ConPaaS is the Platform-as-a-Service component of the Contrail E.U. project.

ConPaaS aims at facilitating the deployment of applications in the cloud. It provides a number of services to address common developer needs. Each service is self-managed and elastic:

- it can deploy itself on the cloud
- it monitors its own performance
- it can increase or decrease its processing capacity by dynamically
  (de-)provisioning instances of itself in the cloud

Copyright ©2011-2012 Contrail consortium.
All rights reserved.

Register

username

email

password

retype password

first name

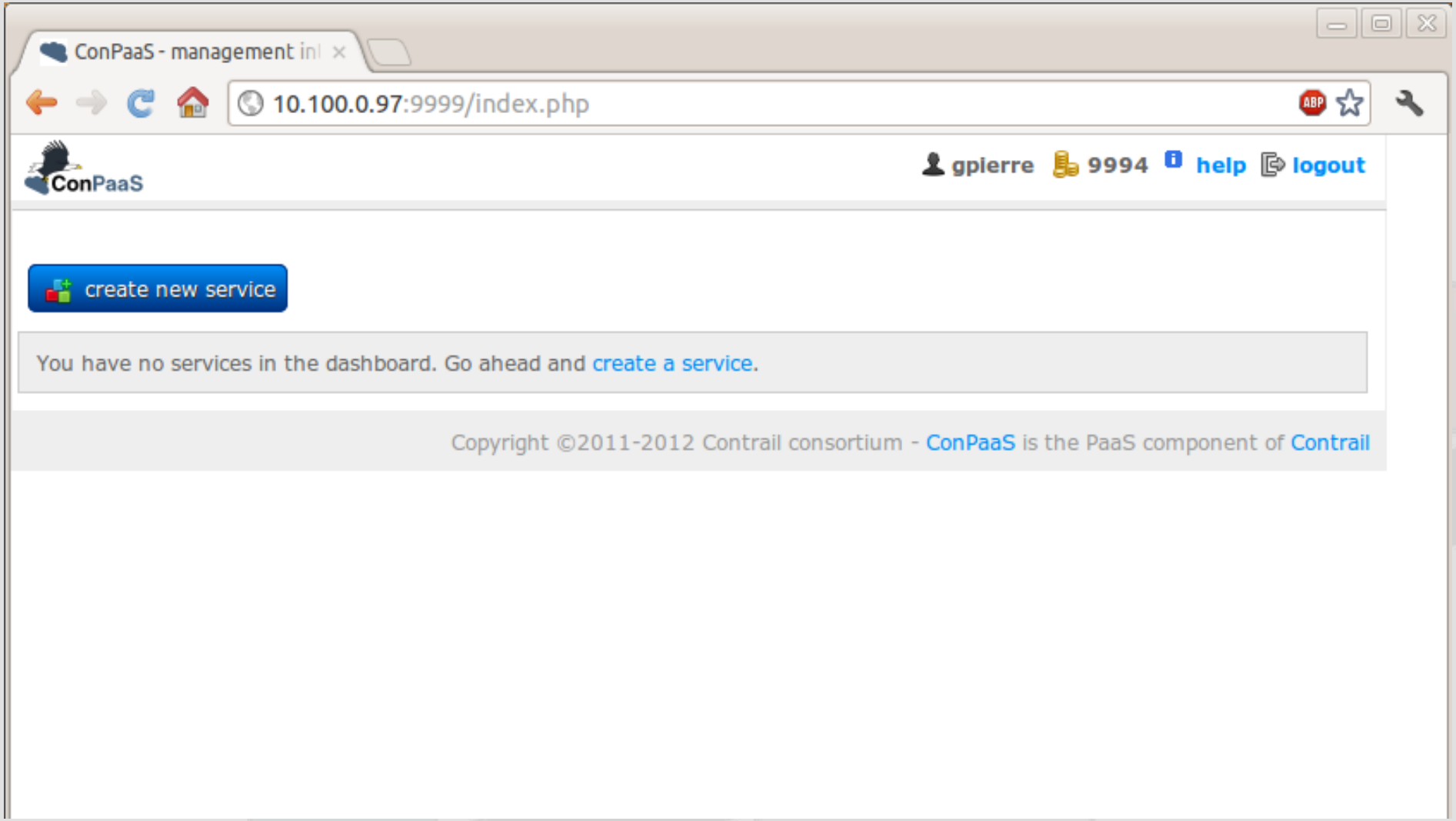
last name

affiliation

Type the two words:

reCAPTCHA™ stop spam. read books.

The ConPaaS Front-End



The ConPaaS Front-End

ConPaaS - create new serv

10.100.0.97:9999/create.php

ConPaaS

gpierre 9999 help logout

please select one of the services below

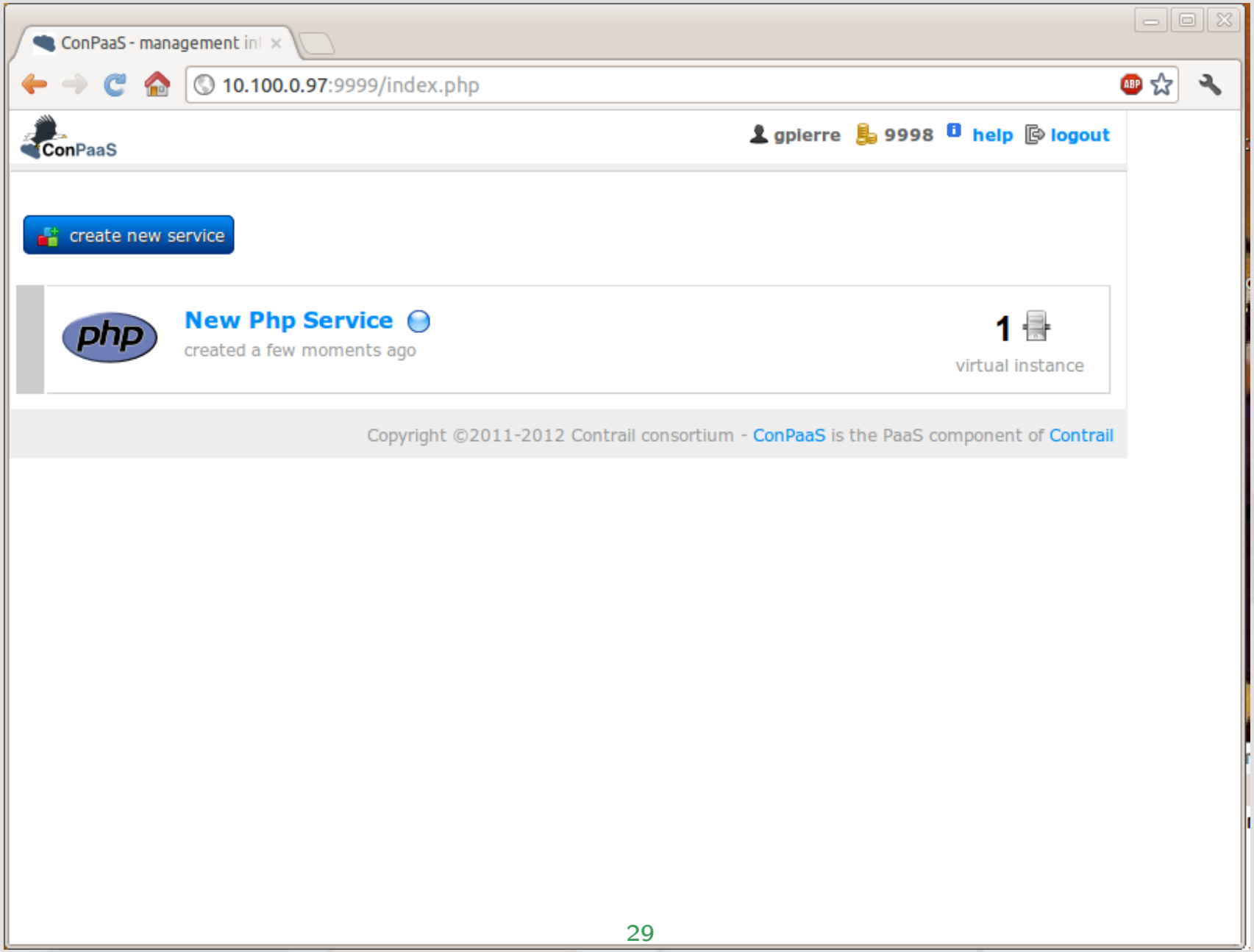
php	**php**	PHP version 5.2 under Nginx
	java	Java Servlet container using Apache Tomcat 7.2
MySQL	**mysql**	MySQL 5.2 Database
Scalarix	**scalarix**	in-memory key-value store
	map-reduce	Hadoop MapReduce cluster
	task farm	Service for running bags of tasks
Se	**selenium**	Selenium functional testing service

cloud provider OpenNebula / Amazon EC2 — Only OpenNebula is enabled on this deployment

create service

javascript: void(0);

The ConPaaS Front-End



The ConPaaS Front-End

ConPaaS - management ini

10.100.0.97:9999/service.php?sid=6

ConPaaS

gpierre 9998 help logout

back to Dashboard

New Php Service start terminate

manager log

initialized · init a few moments ago

1 instance running on OpenNebula

Instance 15751 manager 10.100.0.20

running

Code management

you may update the stage by

uploading archive

or by

checking out repository

Choose File No file chosen

example: .zip, .tar of your source tree

available code versions

code-default code-default.tar • **active** • download a few moments ago

Settings

Software Version 5.3

Maximum script execution time 30 seconds

Memory limit 128M

The ConPaaS Front-End

ConPaaS - management ini

10.100.0.97:9999/service.php?sid=6

ConPaaS

gpierre 9997 help logout

back to Dashboard

New Php Service stop

access application · manager log

running · started a few moments ago

2 instances running on OpenNebula

Instance 15751 manager 10.100.0.20
running

Instance 15752 proxy web php 10.100.0.21
running

add or remove instances to your deployment

0 proxy 0 web 0 php submit

Code management access application

you may update the stage by
uploading archive
or by
checking out repository

Choose File No file chosen
example: .zip, .tar of your source tree

available code versions

code-default code-default.tar · **active** · download a few moments ago

Settings

The ConPaaS Front-End

ConPaaS - management in

10.100.0.97:9999/service.php?sid=6

ConPaaS

gpierre 9997 help logout

back to Dashboard

New Php Service stop

access application · manager log

running · started a few moments ago

2 instances running on OpenNebula

Instance 15751 manager 10.100.0.20
running

Instance 15752 proxy web php 10.100.0.21
running

add or remove instances to your deployment

0 proxy +2 web 0 php submit

The page at 10.100.0.97:9999

no. of instances (e.g. +1, -2)

+2

Cancel OK

Code management

ss application

you may update the stage by
uploading archive
or by
checking out repository

Choose File No file cho

example: .zip, .tar of your source tree

available code versions

code-default code-default.tar · **active** · download a few moments ago

Settings

Software Version 5.3

Maximum script execution time 30 seconds

Memory limit 128M

save

The ConPaaS Front-End

ConPaaS - management in

10.100.0.97:9999/service.php?sid=6

ConPaaS

gpierre 9994 help logout

back to Dashboard

New Php Service stop

access application · manager log

running · started a few moments ago

5 instances running on OpenNebula

Instance 15751 running	manager	10.100.0.20
Instance 15752 running	proxy	10.100.0.21
web		
Instance 15753 *running*		10.100.0.22
Instance 15754 *running*		10.100.0.23
Instance 15755 running	php	10.100.0.24

add or remove instances to your deployment

0 proxy 0 web 0 php submit

Code management access application

you may update the stage by

uploading archive

or by

checking out repository

Choose File No file chosen

example: .zip, .tar of your source tree

available code versions

code-default code-default.tar · **active** · download a few moments ago

Settings

The ConPaaS Front-End

ConPaaS - management in

10.100.0.97:9999/service.php?sid=6

ConPaaS

gpierre 9994 help logout

back to Dashboard

New Php Service stop

access application · manager log

running · started a few moments ago

5 instances running on OpenNebula

Instance 15751 manager running		10.100.0.20
Instance 15752 proxy running		10.100.0.21
web		
Instance 15753 *running*		10.100.0.22
Instance 15754 *running*		10.100.0.23
Instance 15755 php running		10.100.0.24

add or remove instances to your deployment

0 proxy 0 web 0 php submit

Code management access application

you may update the stage by

uploading archive

or by

checking out repository

Choose File No file chosen

example: .zip, .tar of your source tree

available code versions

code-y3gOfl wbas.tar · set active · download	a few moments ago
code-default code-default.tar · **active** · download	a few moments ago

The ConPaaS Front-End