ACTiCLOUD

ACTivating Resource Efficiency and Large Databases in the CLOUD

Concepts and Objectives

Georgios Goumas

goumas@cslab.ece.ntua.gr

ACTiCLOUD INFORMATION

EU H2020 project, Grant Agreement Nº: 732366

Start date: 1 Jan 2017

Duration: 36 months

Partners:

Coordinator: ICCS

OBSERVATIONS THAT MOTIVATED ACTiCLOUD

- Severe resource **misuse** in cloud datacenters

- Misuse meaning:
  - Resource **waste** due to very low server utilization (10-20%)
  - Resource **fragmentation**
  - Resource **unavailability** (or very high pricing) for resource-hungry applications
    - driving use case: *in-memory databases*
  - Resource **contention** due to interference between co-executing applications
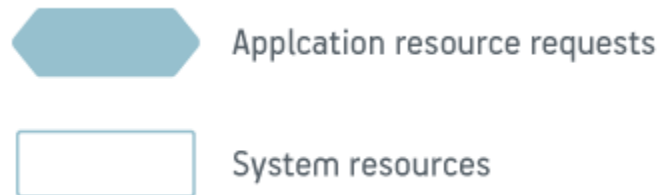
RESOURCE WASTE
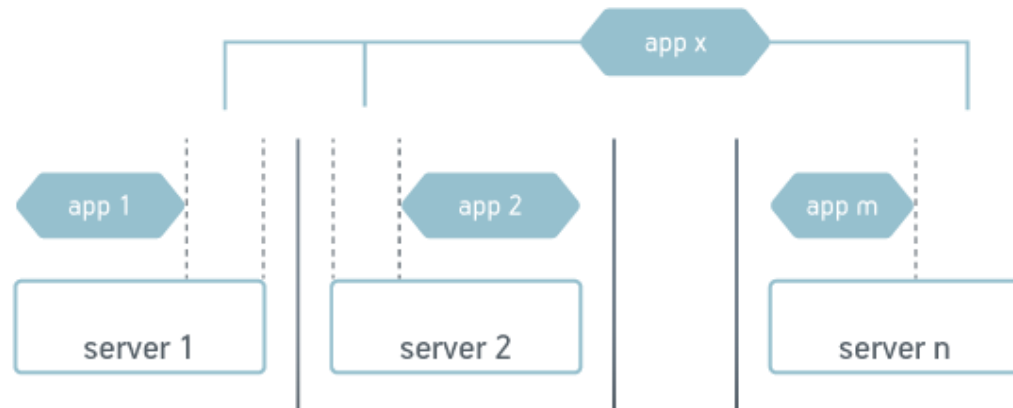
Standby for peak traffic

app 1

app m

server 1   server 2   ...   server n
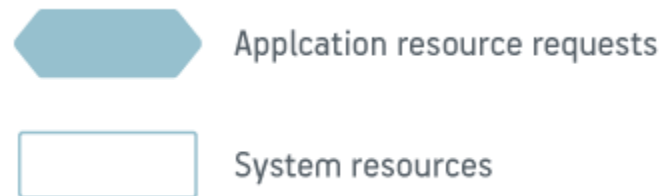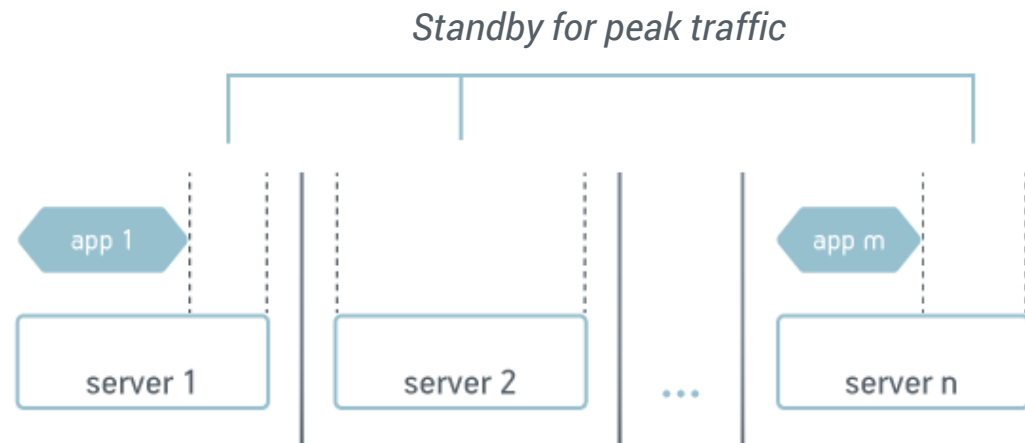
Wasteful allocation by conservative policies to cope with:

- Peak traffic
- Unpredicted collocation behavior

Applcation resource requests

System resources

RESOURCE FRAGMENTATION

Plenty of unallocated resources in small fragments

EnESCE: Workshop on Energy-efficient Servers for Cloud and Edge Computing

RESOURCE UNAVAILABILITY

app x

server 1   server 2   ...   server n

◆ Applcation resource requests

☐ System resources
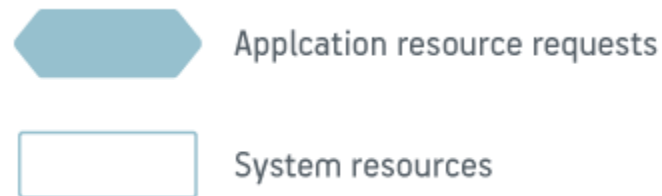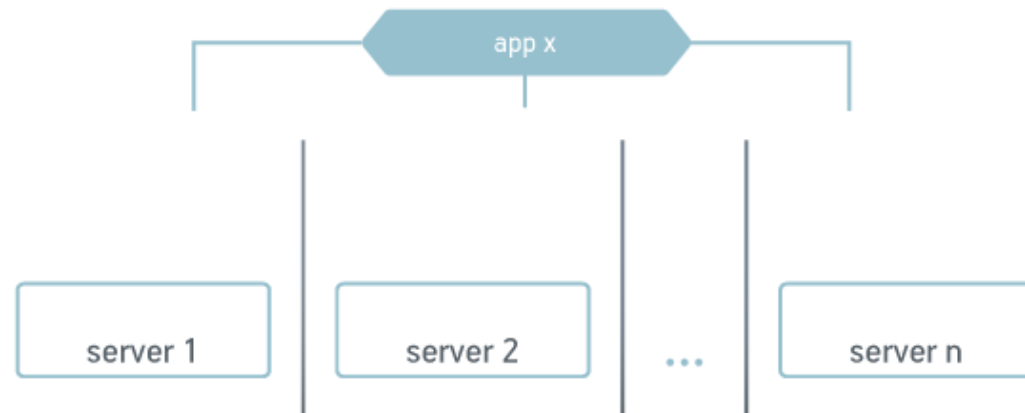
Typical cloud servers cannot service resource-hungry applications

Special application in mind: in-memory databases

We use the term "fat VMs"

Typical scarce resource: main memory

RESOURCE CONTENTION (INTERFERENCE)
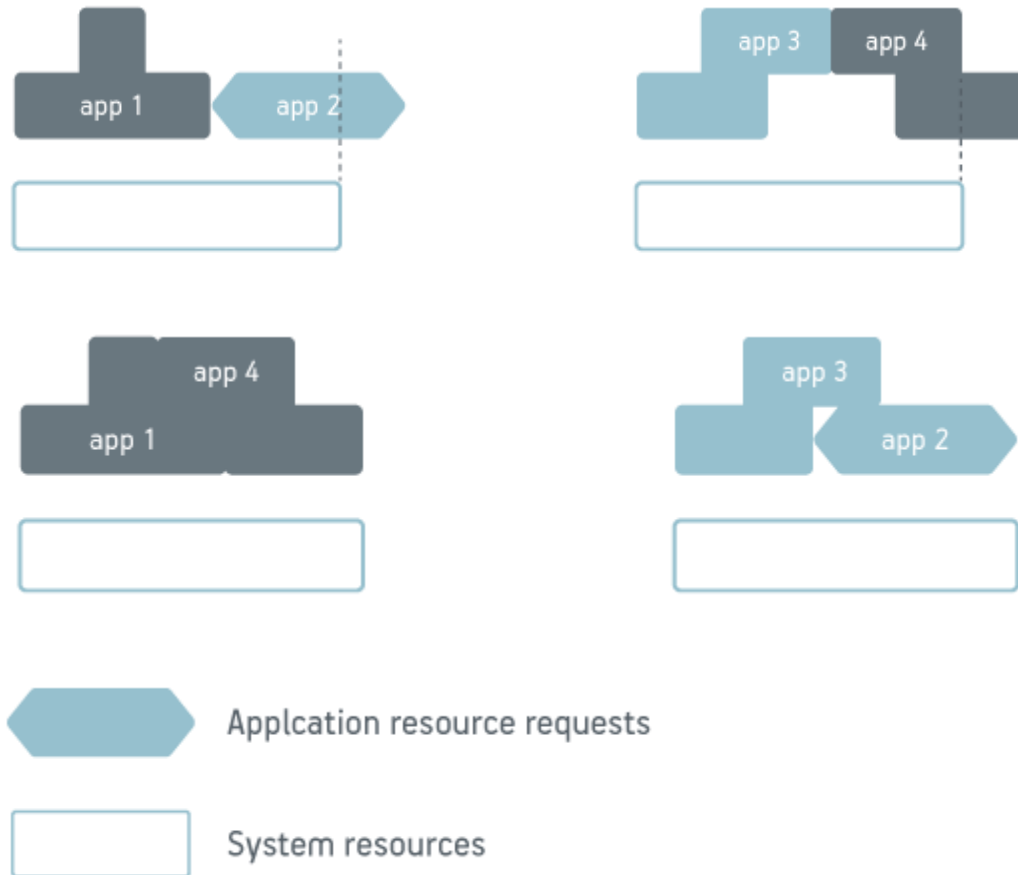
Cloud resource allocators are interference-unaware

Interference can severely degrade performance and QoS

Lack of interference awareness and predictability is a main reason for low server utilization

Applcation resource requests

System resources

ACTiCLOUD STRATEGIC OBJECTIVES

- **Strategic Objective 1 (SO1):** Effective utilization of cloud resources.
  - **SO1.1:** Resource **efficiency**
  - **SO1.2:** Performance **stability**

- **Strategic Objective 2 (SO2):** Deployment of resource demanding applications in the cloud (*special focus on database applications*)
  - **SO2.1: Scalability** in resource provisioning
  - **SO2.2: Elasticity** in resource provisioning

ACTiCLOUD APPROACH

Step1: Break the PC architecture barrier

Base on the technologies provided by Numascale and Kaleao for **resource disaggregation**

server 1 | server 2 .. ... | server k

ACTiCLOUD APPROACH

Step 2: Extend hypervisor technologies to pool resources at the rack scale

Extend OnApp's MicroVisor

- Memory management
- Hyper-converged storage
- Inteconnect optimization
- Fault tolerance

Addresses the problems of resource **fragmentation** and resource **unavailability**

| server 1 | server 2 | ... | server k |

ACTiCLOUD APPROACH

rack-scale hypervisor

server 1	server 2	...	server k

Step 2: Extend hypervisor technologies to pool resources at the rack scale
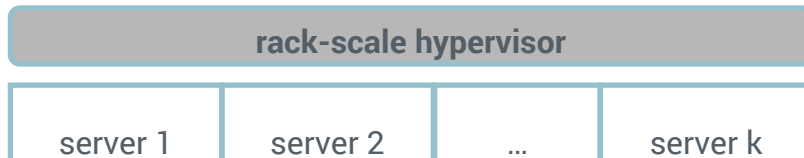
Extend OnApp's MicroVisor
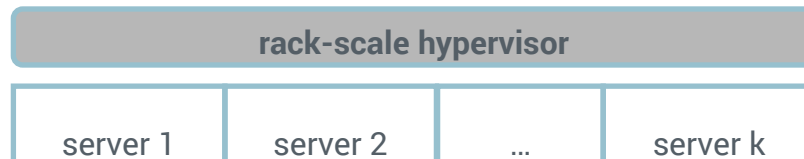
- Memory management
- Hyper-converged storage
- Inteconnect optimization
- Fault tolerance

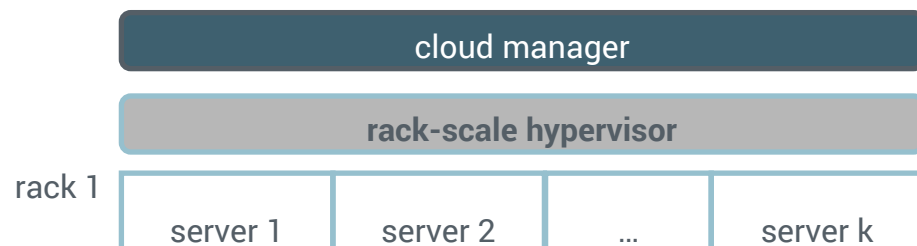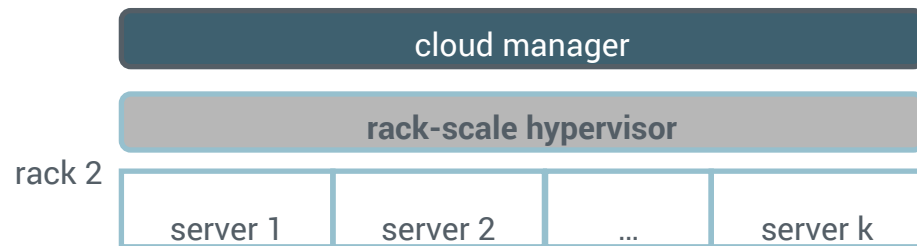Addresses the problems of resource **fragmentation** and resource **unavailability**

ACTiCLOUD APPROACH

Step 3: Enhance the hypervisor and the cloud manager (OpenStack) to handle resources effectively

rack-scale hypervisor			
server 1	server 2	...	server k

ACTiCLOUD APPROACH

Step 3: Enhance the hypervisor and the cloud manager (OpenStack) to handle resources effectively

ACTiCLOUD APPROACH

rack n

cloud manager

rack-scale hypervisor

| server 1 | server 2 | ... | server k |

rack 2

cloud manager

rack-scale hypervisor

| server 1 | server 2 | ... | server k |

rack 1

cloud manager

rack-scale hypervisor

| server 1 | server 2 | ... | server k |

site 1

Step 3: Enhance the hypervisor and the cloud manager (OpenStack) to handle resources effectively

EnESCE: Workshop on Energy-efficient Servers for Cloud and Edge Computing
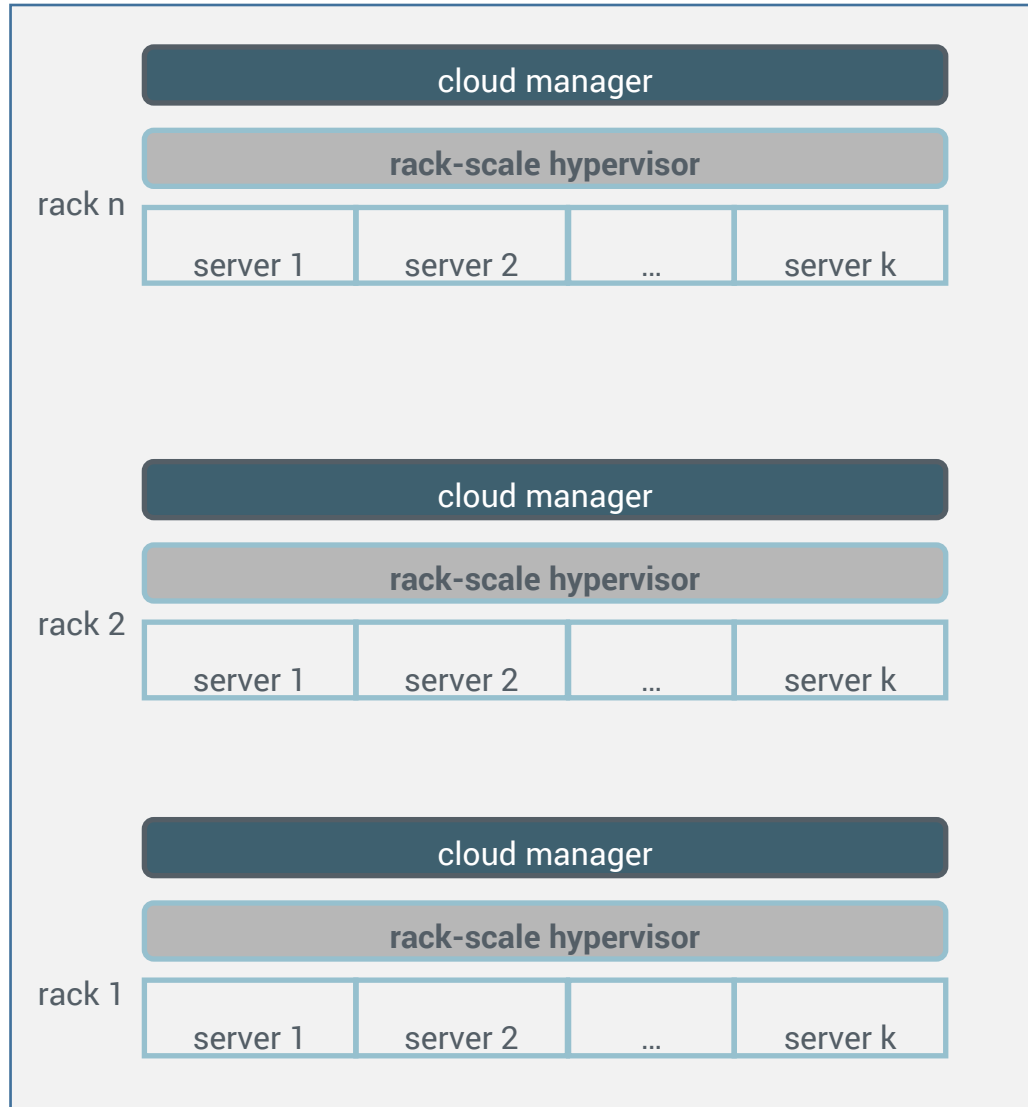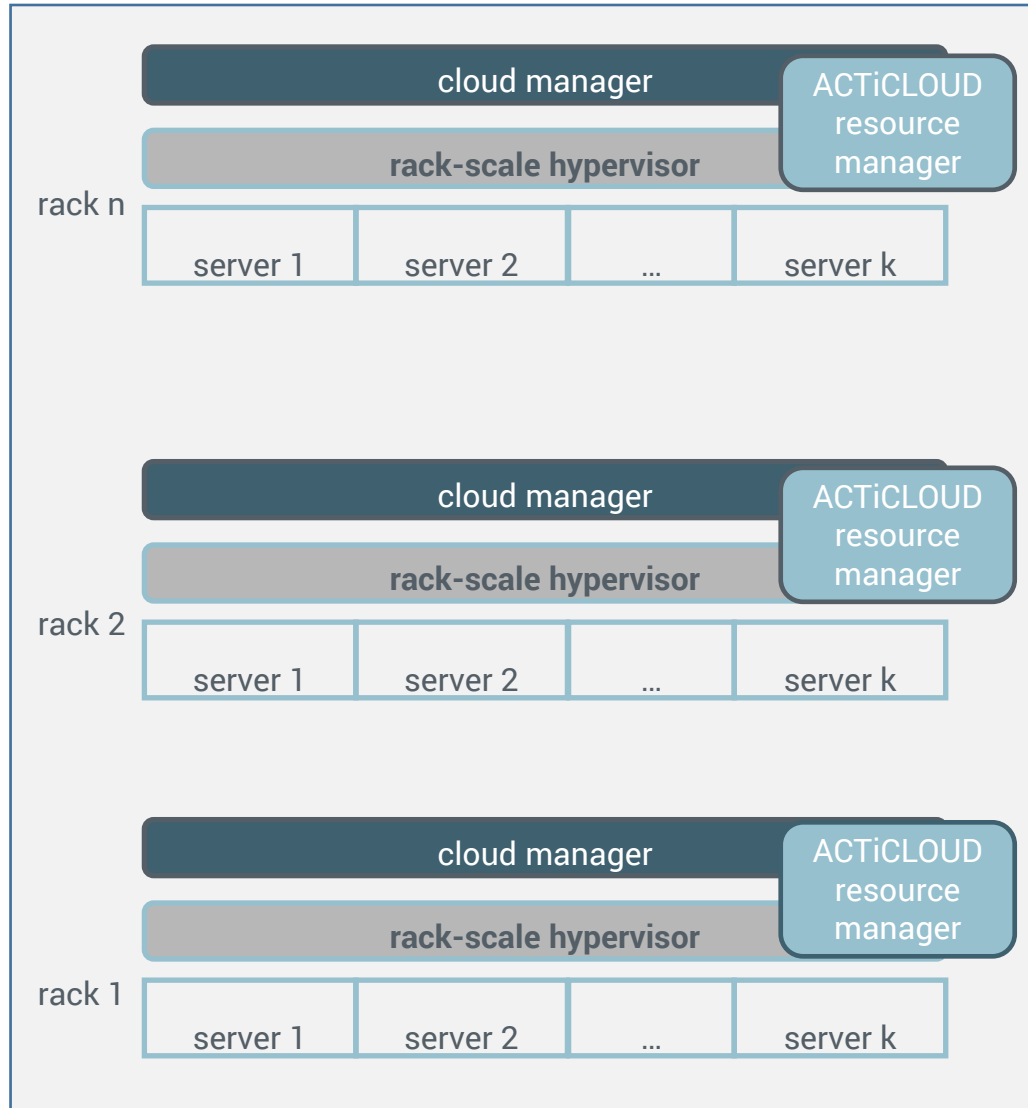
ACTiCLOUD APPROACH

Step 3: Enhance the hypervisor and the cloud manager (OpenStack) to handle resources effectively

ACTiCLOUD APPROACH

Step 3: Enhance the hypervisor and the cloud manager (OpenStack) to handle resources effectively

The ACTiCLOUD resource manager:

- **Monitors** available resources and application requirements

- **Analyzes** the current status and **predicts** the status of alternative states

- **Decides** on alternative resource allocations

- **Takes actions** (e.g. migrate, postpone, time-schedule)

ACTiCLOUD APPROACH

Step 3: Enhance the hypervisor and the cloud manager (OpenStack) to handle resources effectively
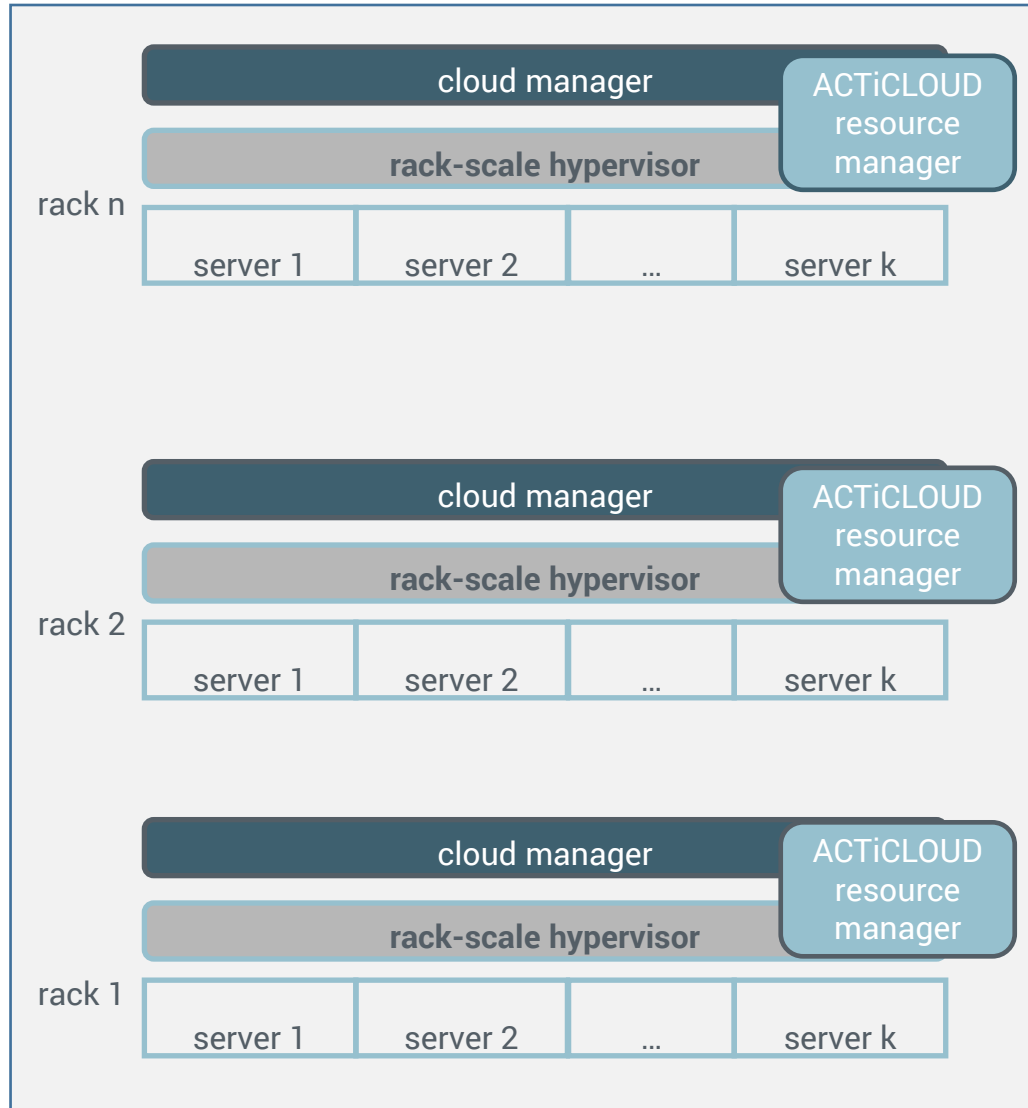
The ACTiCLOUD resource manager operates:

- At the **rack level** (in cooperation with the hypervisor)

ACTiCLOUD APPROACH

rack n

cloud manager

ACTiCLOUD resource manager

rack-scale hypervisor

| server 1 | server 2 | ... | server k |

rack 2

cloud manager

ACTiCLOUD resource manager

rack-scale hypervisor

| server 1 | server 2 | ... | server k |

rack 1

cloud manager

ACTiCLOUD resource manager

rack-scale hypervisor

| server 1 | server 2 | ... | server k |

site 1

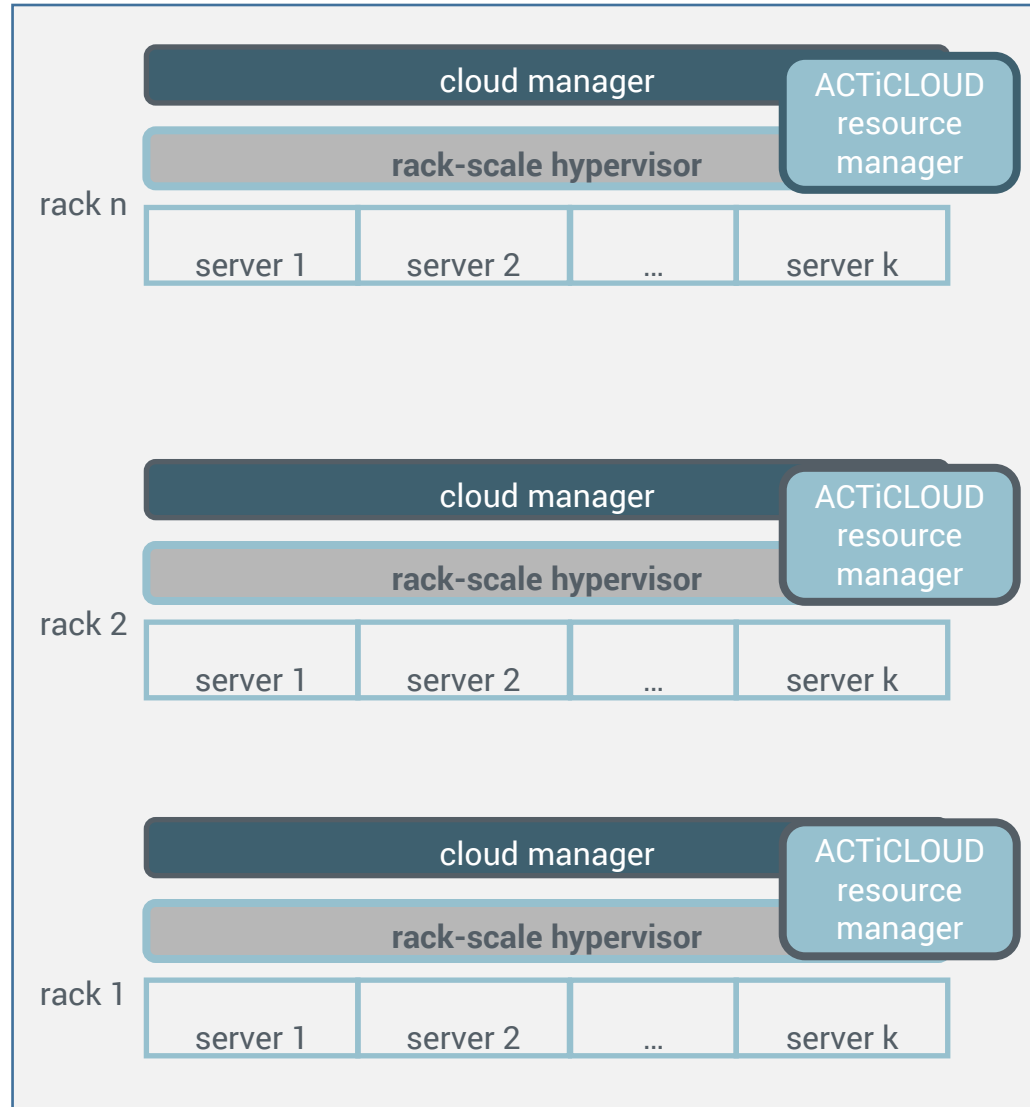**Step 3:** Enhance the hypervisor and the cloud manager (OpenStack) to handle resources effectively

The ACTiCLOUD resource manager operates:

- At the **rack level** (in cooperation with the hypervisor)

- At the **site level** (in cooperation with OpenStack)

ACTiCLOUD APPROACH

rack n

cloud manager | ACTiCLOUD resource manager
rack-scale hypervisor
server 1 | server 2 | ... | server k

rack 2

cloud manager | ACTiCLOUD resource manager
rack-scale hypervisor
server 1 | server 2 | ... | server k

rack 1

cloud manager | ACTiCLOUD resource manager
rack-scale hypervisor
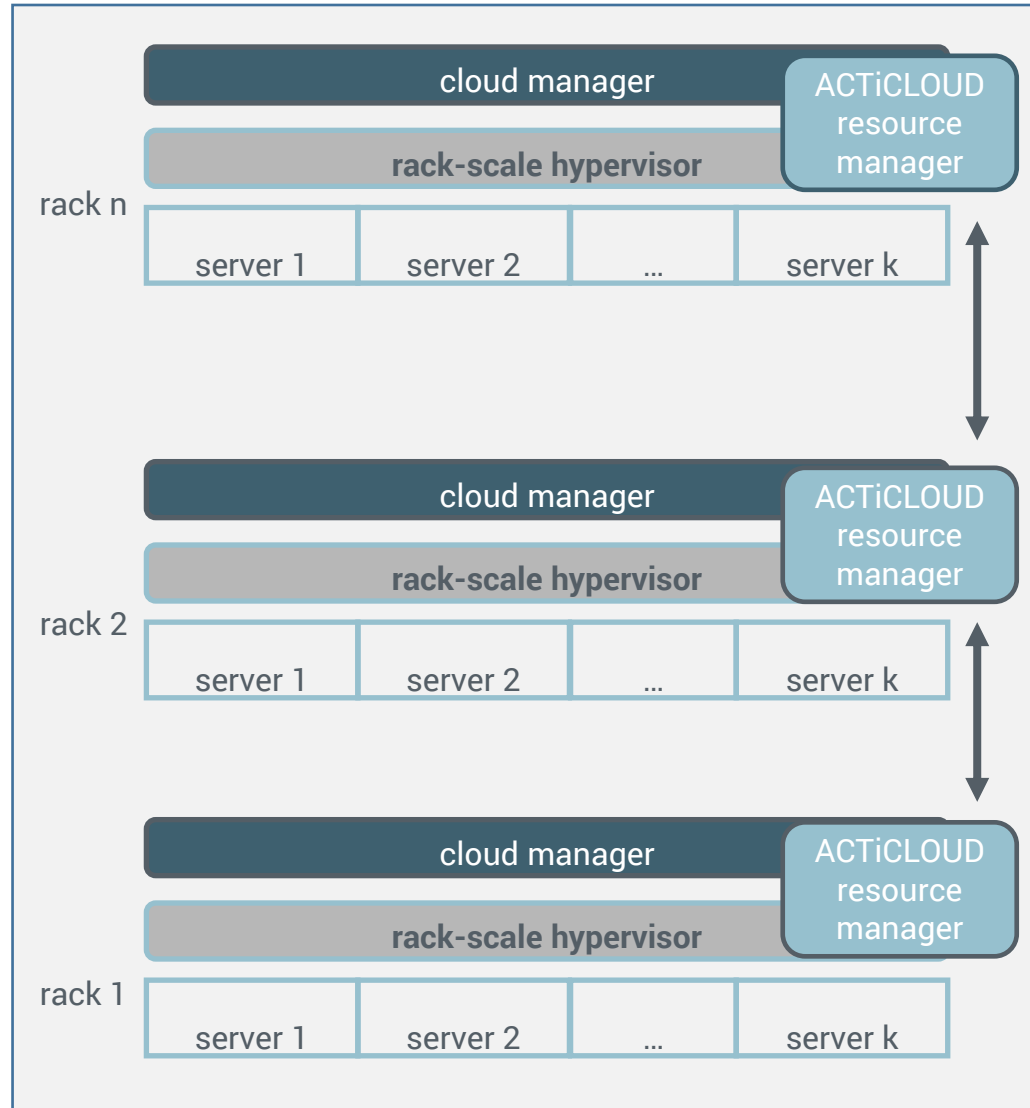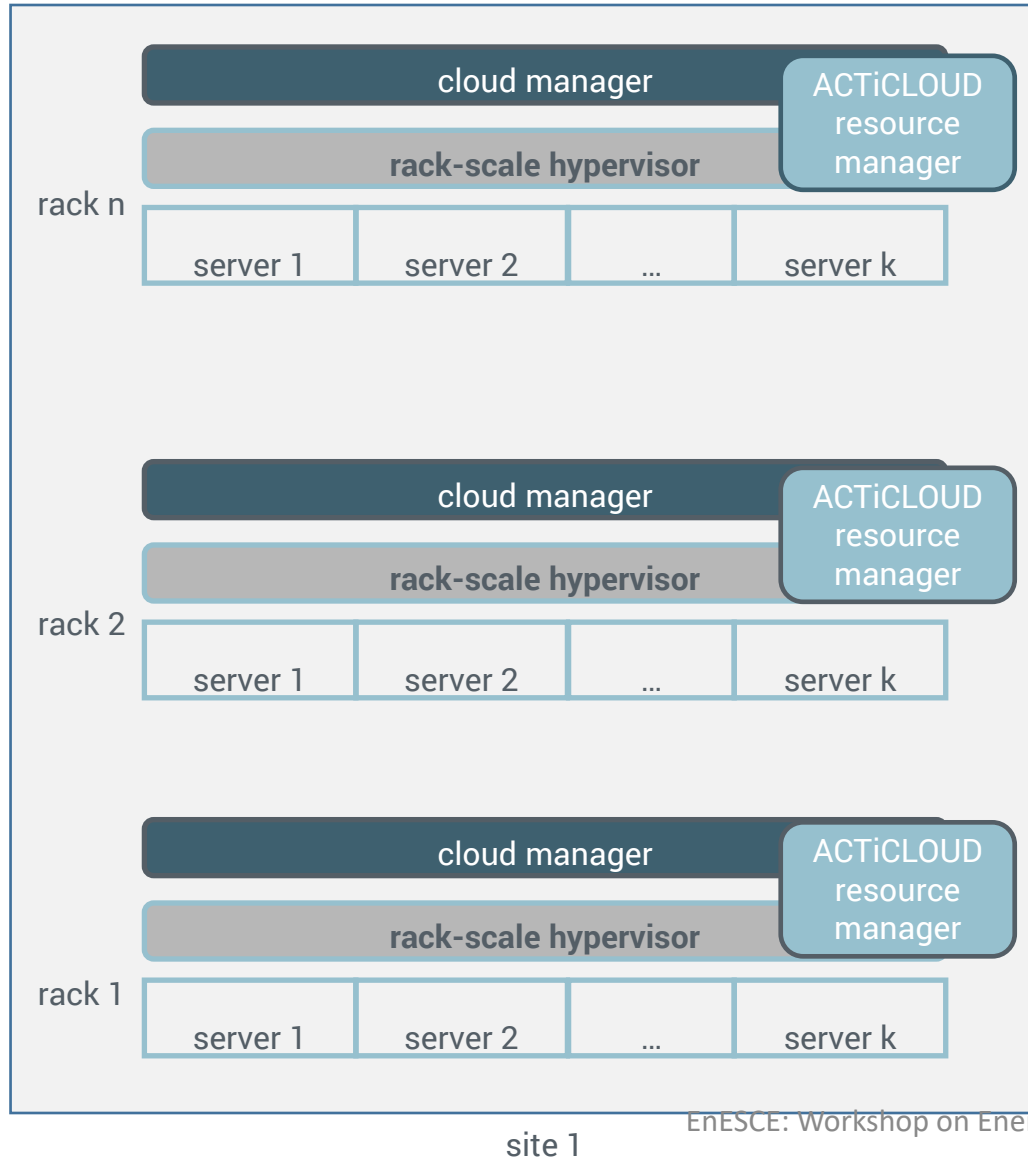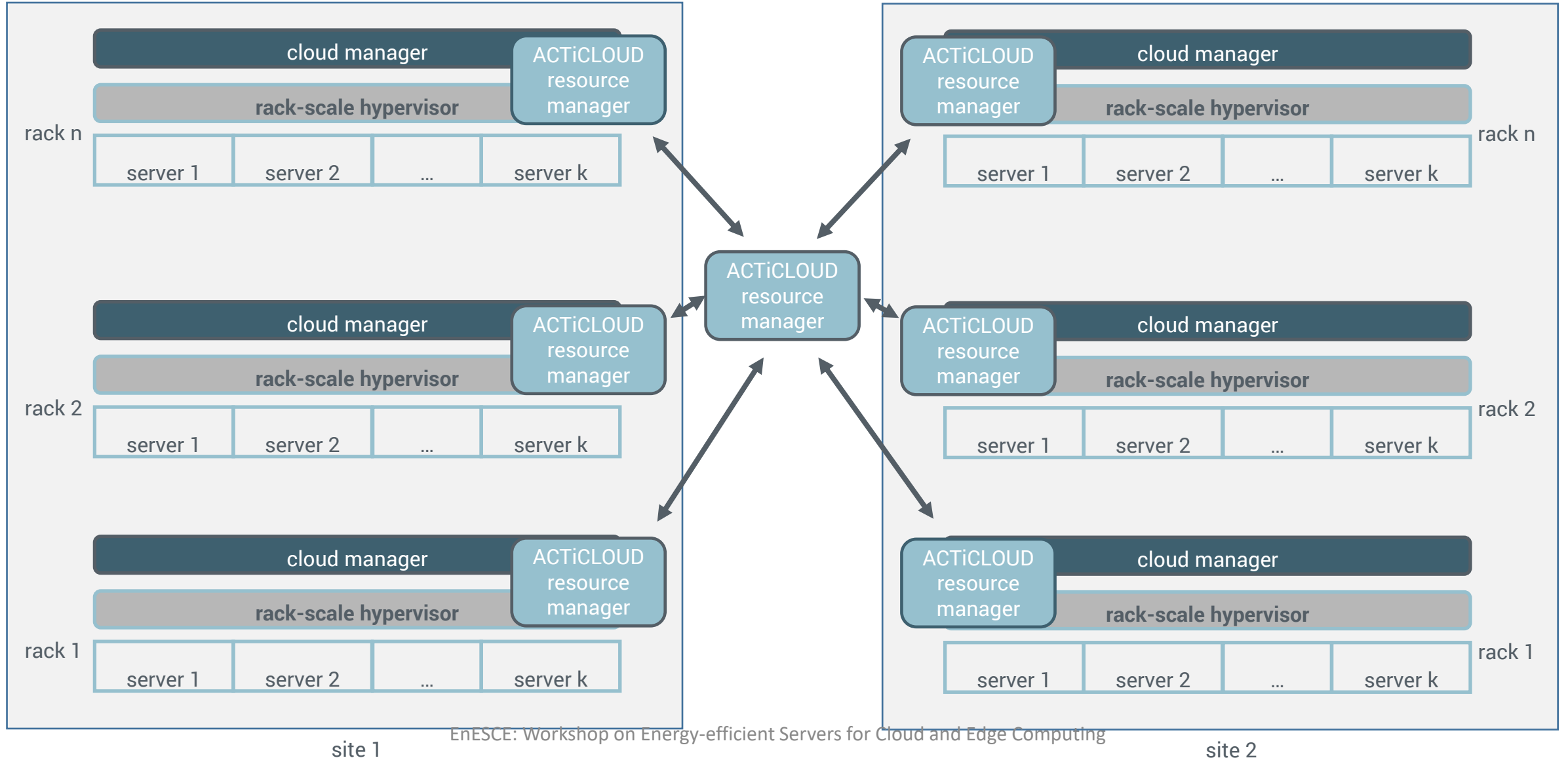server 1 | server 2 | ... | server k

site 1

Step 3: Enhance the hypervisor and the cloud manager (OpenStack) to handle resources effectively
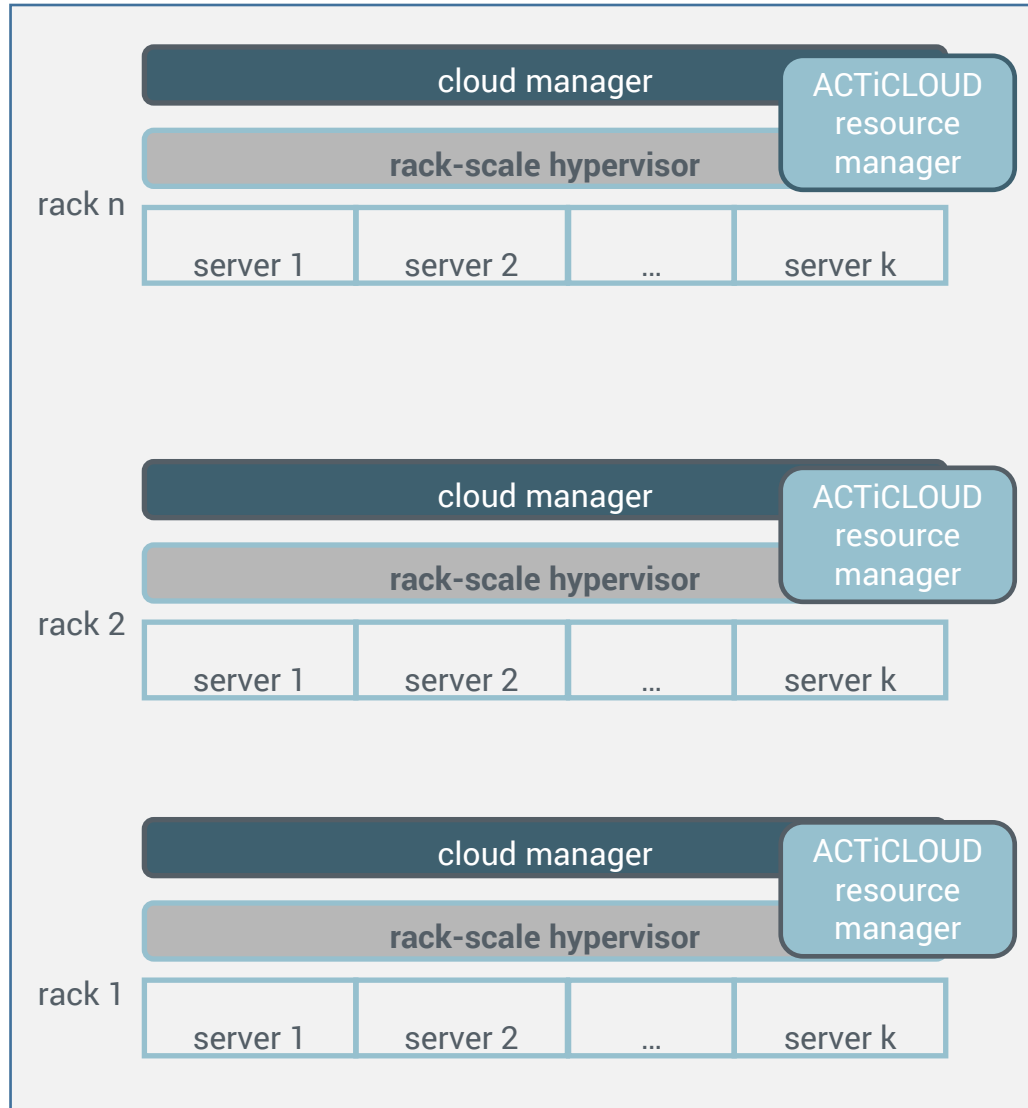
The ACTiCLOUD resource manager operates:

- At the **rack level** (in cooperation with the hypervisor)

- At the **site level** (in cooperation with OpenStack)

- At the **cross-site level** (by utilizing extensions of OpenStack, e.g. Tricircle)

ACTiCLOUD APPROACH

rack n

cloud manager

ACTiCLOUD resource manager

rack-scale hypervisor

| server 1 | server 2 | ... | server k |

rack 2

cloud manager

ACTiCLOUD resource manager

rack-scale hypervisor

| server 1 | server 2 | ... | server k |

rack 1

cloud manager

ACTiCLOUD resource manager

rack-scale hypervisor

| server 1 | server 2 | ... | server k |

ACTiCLOUD resource manager

ACTiCLOUD resource manager

cloud manager

rack-scale hypervisor

| server 1 | server 2 | ... | server k |

rack n

ACTiCLOUD resource manager

cloud manager

rack-scale hypervisor

| server 1 | server 2 | ... | server k |

rack 2

ACTiCLOUD resource manager

cloud manager

rack-scale hypervisor

| server 1 | server 2 | ... | server k |

rack 1

site 1

site 2

EnESCE: Workshop on Energy-efficient Servers for Cloud and Edge Computing

ACTiCLOUD APPROACH

rack n

cloud manager

ACTiCLOUD resource manager

rack-scale hypervisor

| server 1 | server 2 | ... | server k |

rack 2

cloud manager

ACTiCLOUD resource manager

rack-scale hypervisor

| server 1 | server 2 | ... | server k |

rack 1

cloud manager

ACTiCLOUD resource manager

rack-scale hypervisor

| server 1 | server 2 | ... | server k |

site 1

Step 3: Enhance the hypervisor and the cloud manager (OpenStack) to handle resources effectively

Addresses the problems of resource **contention** and resource **waste** (sibling, remote sites can be used to offload peak traffic)

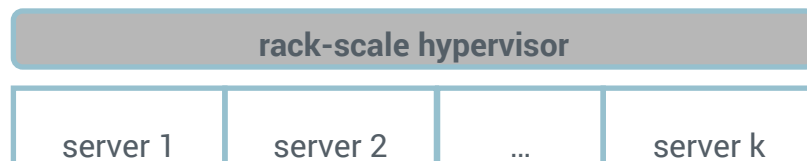# ACTiCLOUD APPROACH: Where we stand

We are half way towards *Activating* **Resource Efficiency** *and* **Large Databases** *in the Cloud*

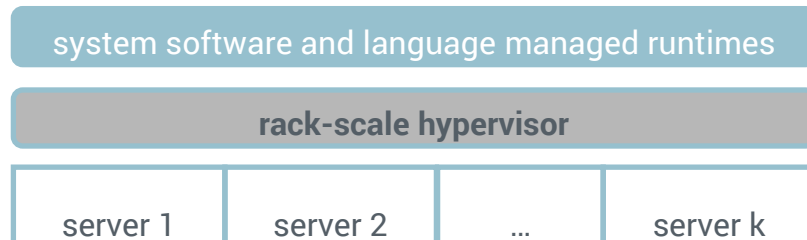What comes next: *Handling large, in-memory databases*

ACTiCLOUD APPROACH

Step 4: Optimize system software and language managed runtimes

Memory managers across the stack

JVM optimizations garbage collector

Other system libraries (e.g. within the host VMs)

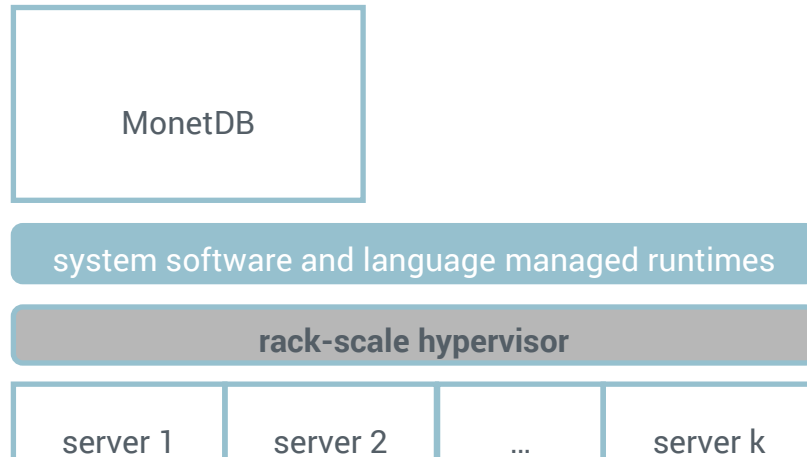| rack-scale hypervisor			
server 1	server 2	…	server k

ACTiCLOUD APPROACH

Step 4: Optimize system software and language managed runtimes

Memory managers across the stack

JVM optimizations garbage collector

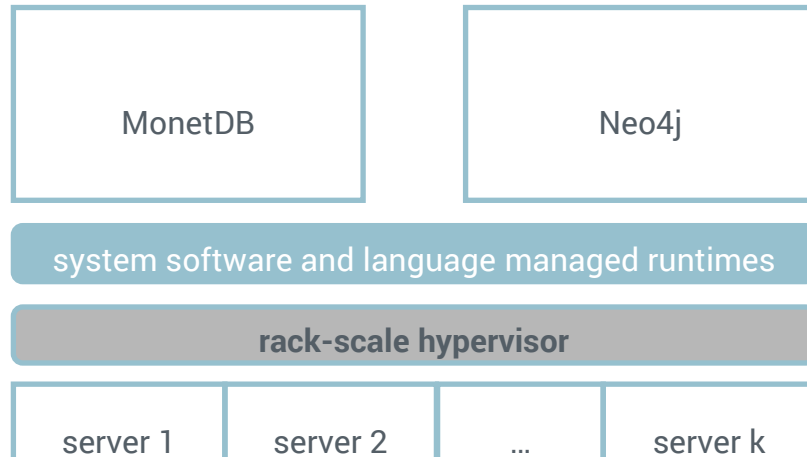Other system libraries (e.g. within the host VMs)

system software and language managed runtimes			
rack-scale hypervisor			
server 1	server 2	…	server k

ACTiCLOUD APPROACH

MonetDB

system software and language managed runtimes

rack-scale hypervisor

| server 1 | server 2 | … | server k |

Step 5: Evolution of in-memory *column-store databases*

ACTiCLOUD APPROACH

Step 6: Evolution of large graph *databases*

MonetDB	Neo4j

system software and language managed runtimes

rack-scale hypervisor

server 1	server 2	…	server k

ARCHITECTURE AT A GLANCE
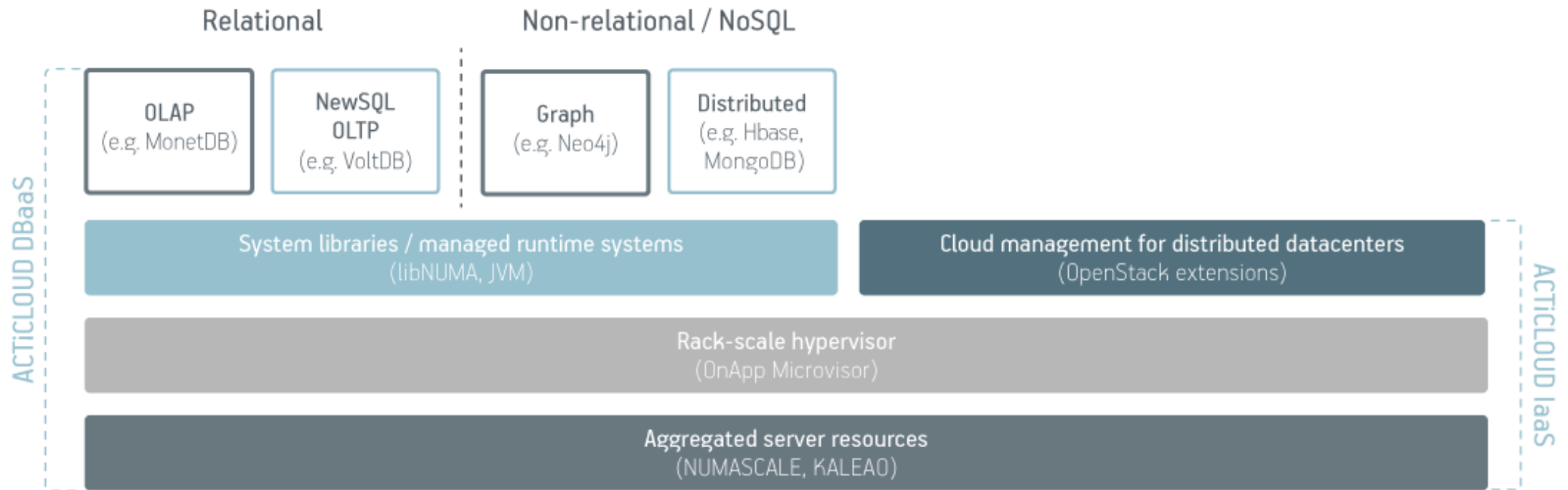
ARCHITECTURE AT A GLANCE - technologies
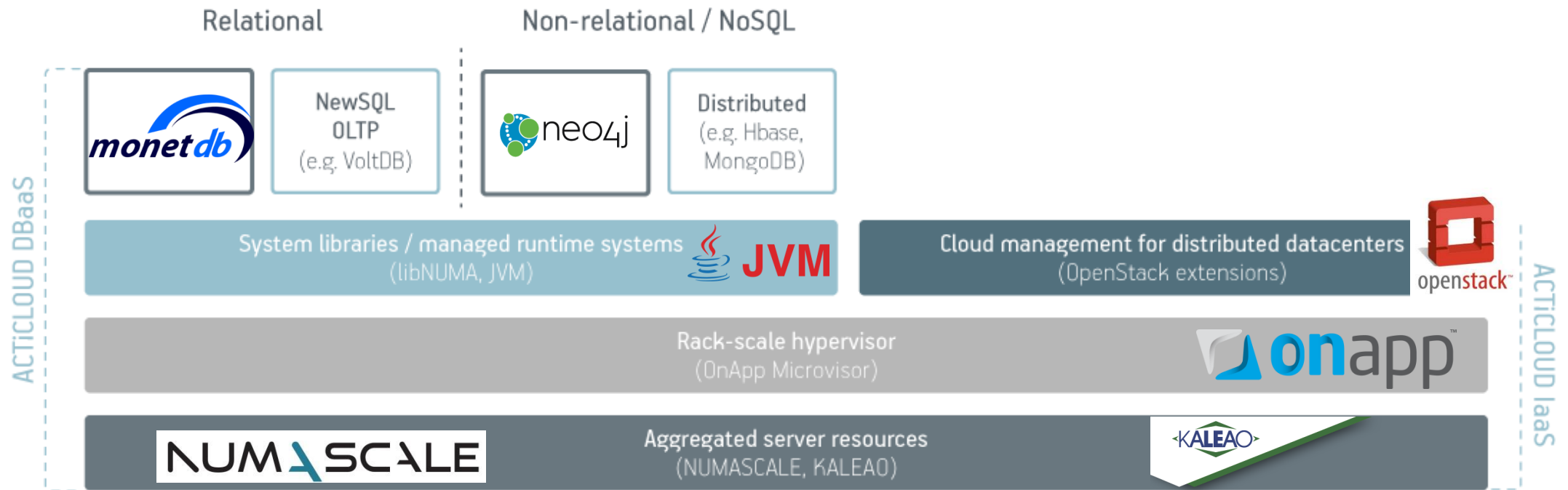
This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement no 732366 (ACTiCLOUD)

Questions?

More info:

- Georgios Goumas: goumas@cslab.ece.ntua.gr

- acticloud.eu

- twitter.com/ACTiCLOUD