Received: 19 October 2015 | Revised: 8 September 2016 | Accepted: 13 October 2016

DOI: 10.1002/cpe.4049

WILEY

SPECIAL ISSUE PAPER

Nu@ge: A container-based cloud computing service federation

Daniel Balouek-Thomert[1,2] | Eddy Caron[1] | Pascal Gallard[3] | Laurent Lefèvre[1]

[1]LIP Laboratory, UMR CNRS - ENS DE LYON - INRIA - UCB LYON 5668, University of LYON, France
[2]Newgeneration-SR Paris, France
[3]NSS Non Stop Systems, Croissy Beaubourg, France

Correspondence
Daniel Balouek-Thomert, LIP Laboratory, UMR CNRS - ENS DE LYON - INRIA - UCB LYON 5668, University of LYON, France
Email: daniel.balouek@ens-lyon.fr

Funding information
FSN (Fund for the Digital Society, BPI France).

Summary

The adoption of cloud computing is still limited by several legal concerns that customers may have, such as data sovereignty. In cloud computing, data can be physically hosted in sensible locations, resulting in a lack of control for companies. In this context, we present the Nu@ge project, which aims at building a federation of container-sized datacenters in the French territory. Nu@ge provides a software stack that enables companies to interconnect independent datacenters in a national mesh. A software architecture is presented and implemented as a federation of small datacenters deployed in France. The proposed architecture enables cooperation between local customized-cloud managers and a federation-wide middleware. It uses monitoring information from facilities and performance indicators from physical servers for managing the system, preventing incidents and considering energy efficiency. Additionally, a prototype of a container-sized datacenter has been validated and patented.

KEYWORDS

cloud federation, energy-aware scheduling, IaaS

1 | INTRODUCTION

Cloud computing represents a significant evolution of information and communications technology in usage and organization. It allows companies to increase their competitiveness by lowering IT infrastructure costs and improving quality of service. This field is an opportuinity to new markets and business models as it is expected to grow up to 29% per year until 2019.[1]

Despite its benefits to users, cloud computing raises several concerns of applications, data storage, and processing. Cloud providers reveal few information about geographical location and the way to process of data and applications. As information converted and stored in binary digital form is subject to the laws of the country in which it is located, several concerns are raised from a legal standpoint. Third-party entities or governments could take control of sensible data, and legal protections may not apply if one's data are located outside her country. Additionally, data from a company could be physically hosted with data from others. This causes security risks as one company may attempt to access data of another.

The Nu@ge consortium* comprises 6 SMEs (small and medium-sized enterprise) and 2 research teams aiming to create cloud technology that is open to heterogeneous hardware and software stacks, spread on a regional network among France and that uses low-energy consumption and ecological datacenters.

This Nu@ge project is a large effort that tackles several challenges. In this work, we describe a federated architecture to provision virtual clusters of resources over the network while providing administrators with a control over data location and quality of service. Our solution is based on innovative container-sized datacenters that enables deployment of a cost-effective and high-performance datacenter environment in any location, meshing regional company-owned datacenters.

This paper is structured as follows. Section 2 introduces the related work. Section 3 presents the Nu@ge architecture and the associated technical choices. The architecture is presented in Section 4 while the prototype is described in Section 5. Section 6 discusses an evaluation of energy efficiency, and Section 7 presents the energy-aware adaptive placement of tasks on server. Section 8 concludes the paper.

2 | BACKGROUND AND RELATED WORK

This section presents related work in the context of cloud providers and existing cloud federation approaches. Then, we detail the notion of modular datacenters and its expected features. Finally, we justify the choice of implementing our own distributed storage solution and the different energy-aware scheduling mechanisms.

*Nu@ge is a research project funded by the FSN (Fund for the Digital Society, BPI France) as part of the *Investissement d'Avenir* program. http://www.nuage-france.fr

TABLE 1 Characteristics and availability of the TIA-942 Tier system

	Characteristics of the site infrastructure design topology	Theoretical availability, %
Tier 1	Single path for power and cooling distribution	99.671
	No redundant components	
Tier 2	Single path for power and cooling distribution	99.741
	Includes redundant components	
Tier 3	Multiple power and cooling distribution paths	99.982
	Only 1 active path	
	Includes redundant components	
	Concurrently maintainable	
Tier 4	Multiple power and cooling distribution paths	99.995
	All paths are active	
	Includes redundant components	
	Concurrently maintainable	
	Fault tolerant	

2.1 | Cloud providers

There exists several surveys on cloud providers.[2,3] Most cloud providers operate according to their own models and protocols. It constitutes a problem that cab lead to vendor lock-in and restrict the transition and interoperability across providers.[4] Furthermore, the headquarters and datacenters location show that most providers are based in the United States while only a few are based in Europe.[5] This constitutes one of Nu@ge's motivation, which aims to provide data storage and cloud services in France.

A means to avoid vendor lock-in is to use open IaaS stack such as OpenStack or VMWare vCloud,[6] for creating and managing infrastructure cloud services in private, public, and hybrid clouds.

2.2 | Cloud federation approach

The Cloud federation approach[7] aims to resolve issues of both providing a unified platform for managing resources at different levels and abstracting interaction models of different cloud providers. Several European projects are providing stacks and/or adaptation of cloud-based systems at IaaS levels. Contrail[8,9] aims at solving the vendor lock-in problem by allowing the seamless switch among cloud providers. InterCloud[7] is a federated cloud computing environment that aims at provisioning application in a scalable computing environment, achieving QoS under variable workload, resource and network conditions. In the Reservoir project,[10] the authors propose an architecture for an open federated cloud computing platform. In such architecture, each resource provider is an autonomous entity with its own business goals. Celesti et al[11] proposes the dynamic cloud collaboration, an approach for setting up highly dynamic cloud federations. A distributed agreement must be reached among the already federated partners to dynamically federate a new provider.

2.3 | Modular datacenter

Clouds depend on datacenters, large facilities used to house computer systems and associated components, such as telecommunications and storage systems. A modular datacenter system is a portable method of deploying data center capacity. As an alternative to the traditional data center, a modular datacenter can be placed wherever data capacity is needed.

Modular datacenter systems consist of purpose-engineered modules and components to offer scalable datacenter capacity with multiple power and cooling options. Numerous manufacturers such as Google, IBM , Sun, Verrari, or HP built modular datacenters into standard intermodal containers (shipping containers) with the following key features:

High density: maximum accommodation of servers, storage, and network equipments within a limited surface.

Cost reduction: by comparison to the building and exploitation of a traditional raised-floor datacenter.

Self-contained cooling: self-contained cooling technologies, which can enable a cost savings and improve system reliability.

Environmentally responsible: minimal carbon footprint.

Disaster recovery and aecurity: characterized by the time of autonomy of the container and the physical equipments dedicated to ensure its integrity.

Fast deployment: usually expected to be less than 6 months to be put in service after order to the manufacturer.

Industry relies on the TIA-942 specification[12] to classify the minimum requirements for telecommunications infrastructure of datacenters and computer rooms into 4 categories, presented in Table 1.

2.4 | Distributed storage

As explained later, the Nu@ge project requires resiliency. In case of the loss of connectivity of a datacenter, the data storage must be distributed among the federation while traceability, integrity, and security of data must be ensured. Additionally, the storage system must keep a journal of data modifications to retrieve a coherent state after an incident. The following part evaluates existing distributed storage solutions with the purpose of integrating one suiting Nu@ge needs.

There are 2 main categories of storage,[13,14] Network attached storage (NAS) and storage area network (SAN). The NAS is a file-level computer data storage server connected to a computer network providing data access to a heterogeneous group of clients, while SAN is a dedicated network that provides access to consolidated, block-level data storage.

2.4.1 | Network attached storage

Although the Ceph project[15] is quite close from our requirements, a cluster can only handle 1 file system, which is a serious technical restriction.

The HDFS[16] is conceived to distribute computations between several nodes. One of the nodes, the *namenode* is a necessary gateway to the system. It constitutes a serious bottleneck and is inappropriate for Nu@ge architecture.

GlusterFS, MooseFS, Pohmelfs, and XtreemFS presented various limitations. Unstability issues, troubles with operating system support, or lack of contribution support led us to exclude those projects from our choices.

2.4.2 | Storage area network

Despite its lack of journaling support, Ceph project[15] features an extensive block data storage. Nevertheless, Ceph cluster gives no information about data location. In this context, data traceability, one of the main objective of Nu@ge, could be achieved only by creating a Ceph cluster per datacenter. This solution is not worth considering because of the high resource consumption of Ceph. The Sheepdog initative[17] seems relatively inactive and only works with QEMU/KVM virtualization technologies. Some of Sheepdog technical choices would lead to scalability problems of storage or number of datacenters.

Because no project provided both means to specify data location and journaling support, we decide to initiate a new project over an SAN, as it is less complex to implement. Unlike an NAS that needs the installation of a software on the client desktop, block-level data storage can be access through standard protocols (specifically iSCSI,[18] supported in a native fashion by numerous operating systems).

2.5 | Energy-aware scheduling

Despite the increasing popularity of cloud computing, infrastructures on which they rely are seldom fully used,[19] mostly because of over-provisioning to handle peak demands. Workloads with large variations in demand can lead to periods of low resource use. As resources are generally not energy proportional, meaning their power consumption at low load is already high, the energy efficiency of an infrastructure is reduced during such periods. Power saving techniques proposed to circumvent such problems consist in slowing down certain server components[20,21] during periods of light load—techniques that according to Le Sueur et al are becoming less attractive on modern hardware[22]—or using software techniques to put idle servers into low-power consumption modes (including shutdown states).[23,24] These techniques are well suited to clouds where virtualization is mainstream.

Nu@ge provides techniques for assigning virtual machines to federated resources, by exploring energy-efficient resource provisioning. We defined mechanisms to adapt resource allocation according to energy-related events and administrator-defined rules.[25]

3 | THE NU@GE ARCHITECTURE

Nu@ge defines a software stack as a coherent set of tools to homogenize management and exploitation of the resources. This section describes the Nu@ge architecture and its main components.

3.1 | Overview

The architecture of the Nu@ge project addresses several system administration concerns, namely, providing a single and shared vision of the whole infrastructure; simplifying service implementation; and managing virtual clusters and associated QoS. Nu@ge aims to virtualize any service. This choice breaks the link between logical resources and physical resources. In particular, we consider only the QoS of virtual/logical resources ignoring the underlying hardware. Nu@ge is modular and favors the autonomy of each component. In this context, a virtual resource can be migrated depending on the following circumstances:

- hardware failures;
- performance optimization;
- energy efficiency improvement; and
- respect of QoS constraint.

A rack, the unit of administration contains:

- equipments dedicated to virtualization, called V-nodes;
- equipments dedicated to storage, called storage nodes;
- network equipments dedicated to internal communications within the rack;
- network equipments dedicated to communication with other datacenters; and
- electrical equipments allowing the supervision and interventions.

The high-level components and their features are described in the following sections.

3.2 | V-node

A V-node is a physical node dedicated to the execution of virtual systems. Several infrastructures services are required, including

- interconnection between Nu@ge and the various IaaS providers,
- setting up of network services, and
- piloting process of electrical alimentations

The main virtual machines deployed in the system are as follows:

Internet gateway: provides Internet access to nodes, physical, or virtual, present in the Nu@ge infrastructure. This machine enables the creation of filtering rules (firewall) to set a first level of security for network services.

VPN gateway: offers a secure access to Nu@ge's internal resources. Identification, authentication, and data encryption are performed with digital certificates, which are created and managed individually for each Nu@ge user.

IaaS gateway: this is the component that links Nu@ge to the IaaS platform for the end user. This virtual equipment is the separation between Nu@ge's area of responsibility and the IaaS administrators.

***DNS service*:** the DNS is a primary service of Internet enabling the resolution of identifiers, required for Internet browsing.

Storage access service: creates storage units dynamically for the IaaS platforms. The storage units are available as file systems or hard drives. This service is linked to an IaaS exposing a dedicated storage zone to the Nu@ge infrastructure.

3.3 | Storage node

The main objectives of the distributed storage system are availability, traceability, and integrity and safety.

For each IaaS hosted in the Nu@ge architecture, a storage cluster is created. The number and the location of hosts depend on the contract established with the IaaS owner.

Storage nodes are machines with significant storage resources. High-performance disks allow improved writing/reading operations while traditional disks offer larger storage capacity with greater access time and latency.

Storage nodes are connected using a dedicated subnetwork, as they need to securely exchange user's data. For that purpose, the nodes have 2 Gbits/Ethernet interfaces and an InfiniBand interface. The QoS is guaranteed, in particular during data replication, to ensure resiliency in case a datacenter is suddenly not available. Additionally, the system keeps a journal of data modifications.

Unlike V-nodes, a storage node provides locally to the nearby V-nodes storage resources. A storage node has a high-storage capacity sets of hard drives, each set containing dozens of hard drives.

3.4 | Network infrastructure

We use 2 kinds of networks within the Nu@ge architecture: internal, dedicated to the communication between the different IaaS and external, used for the interconnection with end users.

The internal network allows the creation of private networks between a user's nodes. Private networks require an IP addressing intra- and inter-datacenter, in which the flows of information are encapsulated. As the interconnection with end user is performed via third-party Internet providers, it is necessary to have several networks, depending on the segmentation set by the internet providers.

3.4.1 | External communication between the datacenters

A simple method would consist of a star network topology, built around a central site with a full redundancy among the links. In a star topology, every node is connected to a central node called a switch. The switch acts as a server and the peripherals as clients.[26] However, for reasons of cost and architecture consistency, we do not consider that solution.

Ensuring continuity of service, without a star network topology, requires a number of links superior to the number of Nu@ge datacenters. Without any protocol, the interconnection of those links would cause a loop and prevents the delivery of packets.

Spanning tree protocol (STP) is a level 2 protocol (Ethernet) allowing the construction of an Ethernet network without loop.[†] The STP presents a simple approach of the problem by cutting some links, to obtain a tree architecture. Due to its simple functioning, STP is widely used despite a few limitations such as the poor repartition of flows and a convergence time up to 30 seconds.

While several extensions to STP address those limitations, a new protocol named transparent interconnection of lots of links (TRILL) is gaining popularity. The TRILL is an IETF standard.[‡] This protocol presents the avantages of the routers and the network bridges by creating a level 2 network on the different links available.

Then, the protocol sets dynamic routing tables with MAC addresses. Using this level 2 routing, the protocol ensures to always have the shortest path to route packets. In the context of Nu@ge, we use TRILL to manage Ethernet segmentation.

3.4.2 | Virtual machine mobility:

In a context of user mobility and network virtualization, getting a proper identification of an end user over the network can be a difficult task because of the various possibilities of Internet access. The protocol LISP (Locator/ID Separation Protocol) tackles this problem by enabling migration over network while maintaining the same IP address. The LISP is a protocol where IP addresses have 2 roles, namely, localization and identification.[§] The LISP aims to solve problems related to the growing size of IPv4 routing tables. Additionally, the protocol enables users to break the link with a single Internet access provider (mobile users). The LISP addresses this issue by separating the location from the identification. An IP address is used in 2 ways:

- identify a machine present in a network, and
- locate the identifier of the machine to route the traffic in an IP network.

A distributed table of matches allows to find a locator, RLOC (Routing LOCator) from an identifier EID (Endpoint Identifier). The LISP is independent of the IP version and can be deployed in an incremental fashion, without the necessity of having the full Internet architecture supporting it.

4 | REALIZING THE ARCHITECTURE WITH OPEN COMPONENTS

In this section, we describe how we leverage OpenStack and DIET Cloud for realizing the Nu@ge federation architecture. As we consider the datacenter as a complete resource (just like memory, storage, CPU, or network), its management can be integrated to the conception and exploitation of cloud. We use DIET Cloud, an extension of the DIET middleware to collect information from different IaaS and perform federation-wide decisions.

4.1 | OpenStack

OpenStack is an open-source cloud computing platform for both private and public clouds. The OpenStack project was announced in July

[†] STP is defined in IEEE 802.1d-2004

[‡] TRILL is defined in the RFC 6325.

[§] LISP is defined in the RFC 6830.

of 2010 by Rackspace and NASA, who made the initial code contributions. The OpenStack software consists of several independently developed components with well-defined APIs. The core component that provides IaaS functionality is OpenStack Compute (also called *Nova*). It handles provisioning and life-cycle management of virtual machines and supports most available hypervisors. *Neutron* is the component for building virtual network topologies that live atop hardware from different vendors. *Cinder* is the block storage, a scalable storage service similar to Amazon S3. *Horizon* is a web-based GUI, primarily for management purposes such as starting/stopping virtual machines and managing user/group configurations. Further components are available such as image service and identity management. The implementation described in this paper is based on the Grizzly release of the OpenStack, and it uses Compute, Neutron, Cinder, and Horizon, which we have extended for our purposes.

In particular, a Block Storage Service was implemented within *Cinder*. Eguan provides a working backend driver for OpenStack's cinder block storage service with high-availability, real-time data replication and history features. OpenStack volumes and snapshots can be hosted on 1 or multiples eguan instances with integrity checks and precise location of the data. The project's source code is available under the Apache 2.0 license. Implementations details are out of the scope of this paper.

4.2 | Federation scheduler using DIET cloud

We rely on DIET,[27] an open-source middleware that enables the execution of applications using tasks that are scheduled on distributed resources using a hierarchy of agents for scalability. The DIET comprises several elements, including

- **Client** application that uses the DIET infrastructure for remote problem solving.
- **Server daemon** (**SED**) that acts as a service provider exposing functionality through a standardized computational service interface. A single SED can offer any number of computational services.
- **Agents**, deployed alone or in a hierarchy, which facilitates service location and interaction between clients and SEDs. Collectively, a hierarchy of agents provides high-level and scalable services such as scheduling and data management. The head of a hierarchy is termed as **master agent** whereas the others are **local agents**.

The steps of the scheduling process are explained below:

1. *Submission of a virtual machine creation request*

   A client issues a request describing a virtual machine template. If none of the datacenter is able to create new instances, a notification is returned to the client.
2. *Propagation of a request*

   The request is propagated through a hierarchy of agents.
3. *Collection of estimation values*

   Each agent computes and gathers its metrics, particularly performance and energy consumption. A reply containing these values is sent back to the scheduler.
4. *Sorting of candidates*

   Once the scheduler retrieves all replies, it proceeds to a sort according to specific criteria. The first ranked node is then elected and notified.
5. *Virtual machine creation*

   The virtual machine is created on the elected node.

The DIET[27] implements many prerequisites, such as service calls, scalable scheduling, and data management. This allows us to implement a cloud middleware with minimal effort.

The aim of the DIET cloud is to provide an architecture that handles a large number of cloud middleware and cloud service providers. Thus, it hides the complexity and heterogeneity of the cloud API layer, thanks to δ-Cloud.[28] The δ-Cloud is a cloud adapter that provides a library that eases the interfacing with different Clouds. δ-Cloud offers a standardized API definition for IaaS Clouds with drivers for a range of different Clouds. It can be seen as a meta-API. The δ-Cloud API is designed as a RESTful web service and comes with client libraries for all major programming languages.

Using this cloud extension, DIET can act as a federation scheduler by benefiting from the different IaaS capabilities and manage virtual machines. Virtual machine management decisions will be taken according to the monitor systems from the underlying datacenters.

The federation (see Figure 1) establishes relationships between the physical infrastructure and its logical behavior by providing developers (administrator) with an abstract layer to implement aggregation and resource ranking on the basis of contextual information such as infrastructure status, users preferences and requirement, and the energy-related external events that can occur over time.

To perform a placement, information on datacenter health status, energy monitoring and capacity must be obtained.

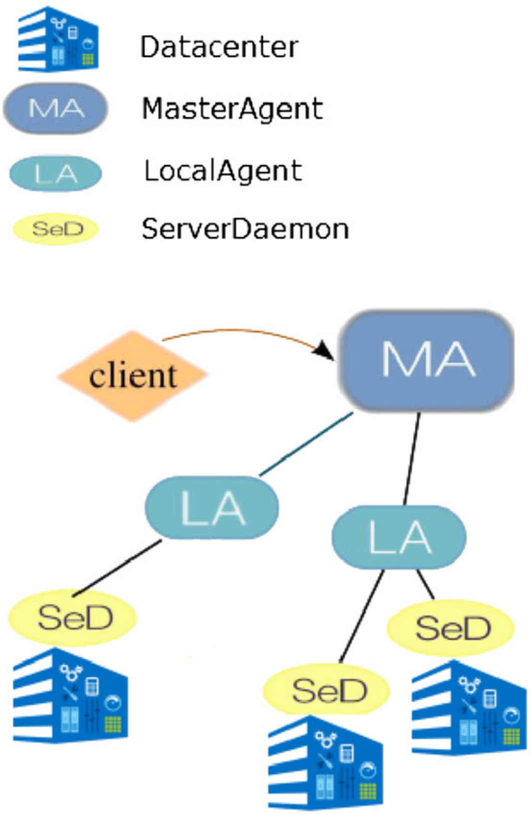


FIGURE 1 Federation scheduling using DIET cloud

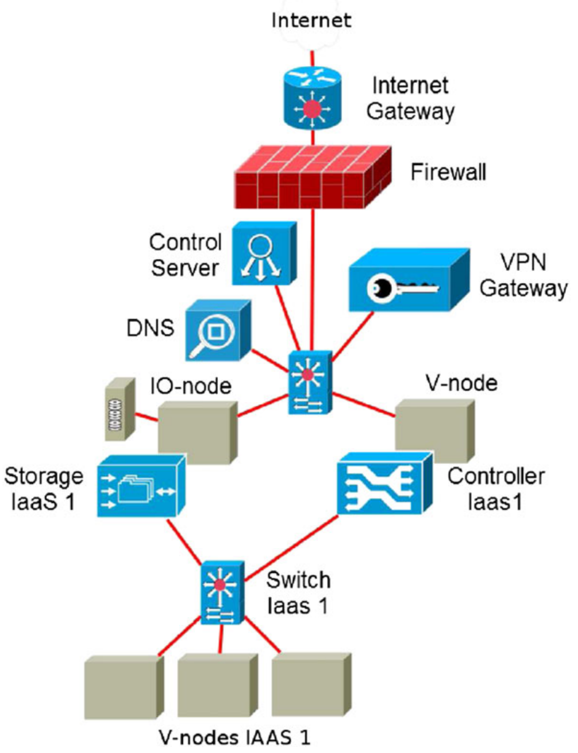


FIGURE 2 Nu@ge architecture including gateways and a IaaS

These metrics are incorporated into DIETSED to populate its estimation vector using new tags. Every time a client submits a request for a virtual machine, each datacenter retrieves its metrics over the local monitoring tools. Once this information is collected, servers are advised to populate and forward an estimation vector to the master agent, which in turn uses an **aggregation method** to sort server responses according to the chosen criteria and select the appropriate resource to execute the client request. Each DIET agent of the hierarchy performs the selection following the plug-in scheduler.

5 | PROTOTYPE

This section briefly describes a prototype implementation of the Nu@ge architecture as depicted in Figure 2. Such implementation has been used for evaluating the performance and feasibility of the proposed approach. The prototype has been deployed and validated over 4 different geographical locations in France (Figure 3).

5.1 | Roles

The IaaS administrator, in charge of the virtual infrastructure offered by Nu@ge, sets and configures resources (eg, physical or virtual nodes, storage disks, and virtual routers).

5.2 | StarDC

The StarDC (Figure 4) features 4 service units of 19-inch racks and can hold up to 168 computing servers. The container provides 15 m^2 of floor space, a power capacity of 18 kW and a power usage effectiveness (PUE) of 1.24. The *StarDC* is built within tier 3 specifications and is the subject of a patent.

Unlike most modular datacenters, the StarDC does not use water cooling. It has a broader range of physical locations and an eco-responsible behavior because free cooling is used to cool the container. StarDC uses a mechanism of temperature using outdoor air as a free cooling source. The purpose is to take advantage of outdoor temperature to naturaly cool of equipments. When the air is injected into machines, its temperature raises by a delta number of 10° (common value among commercialized servers). When the outdoor temperature is higher than a threshold, we use air conditioning to cool it.

The Nu@ge customer is in charge of setting up the cold aisle temperature. If he chooses a temperature of 20° to have a safety margin, the air conditioning will be active approximately 20% of the year (varies depending on the location). Choosing a temperature value up to 25° and more results in less air conditioning and a better ecological impact. We discuss the evaluation of PUE in Section 6.

5.3 | Construction of an IaaS

The creation of a new IaaS does not impact the architecture of Nu@ge. The main changes concerns virtual nodes allowing the sharing of physical resources. In particular, the instantiation of a storage access point; an IaaS access point; and a virtual switch interconnecting the IaaS equipments. As a result, several storage nodes and V-nodes can be used by multiple IaaS.

5.4 | Storage cluster

Nu@ge racks contains 2 storage nodes. As storage management can require large computational resources, a storage node features dual-core CPUs for a total of 24 threads and 256 GB of RAM. Deployment of storage nodes is performed via the following steps:

1. Booting via PXE / TFTP protocols.
2. Configuration using Puppet.
3. Creation and configuration of an object storage in RAID1.
4. Creation and configuration of RAID6 objects.
5. Creation of logical volumes.

Once created, the node executes an Openstack storage service specific to the newly created IaaS, and the storage server. This organization is coherent with Nu@ge objectives of data isolation between IaaS and data traceability for the administrators.

5.5 | Supervision

An interface has been built to visualize information about datacenters and customers. It provides the visualisation of the dynamic mapping of virtual machine deployment on physical infrastructure along with analysis of performance for user activity and alerts related to usage incidents.

This platform acts as an autonomous web board displaying information about a local datacenter and to the global federation. It can be used as a complement or integrated into Openstack's Horizon

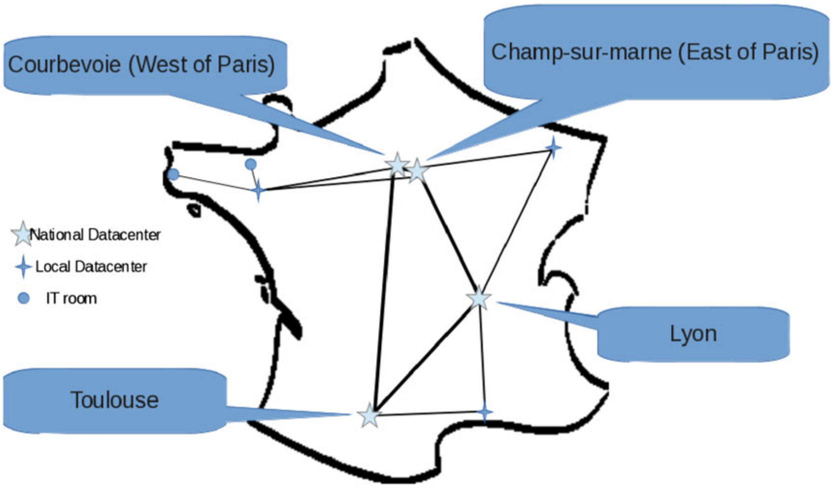


FIGURE 3 Nation-wide deployment over 4 locations in France

FIGURE 4 The public presentation of the StarDC container occured on September 18, 2014, during Nu@ge inauguration in CELESTE headquarters, Marne-la-vallee, France

(Figure 5). Logging has been performed using Nagios Core[29] and SNMP, an Internet-standard protocol for managing devices on IP networks, for nonstandard devices.

6 | PUE OF NU@GE

The PUE is a metric used to evaluate the energy efficiency of a datacenter.[30] From a practical point of view, it measures how much energy is used by the computing equipment in comparison to cooling and other overhead. The PUE is expressed by the ratio:

$$\text{PUE} = \frac{\text{Total Facility Power}}{\text{IT Equipement Power}} \tag{1}$$

Nevertheless, it is very hard to know the real PUE from a company because the area of equipment power can be debatable. As an example, for the Google datacenter, considering only servers and air conditioning give a PUE of 1.06. However, if Google includes generators, transformers, site substations, and natural gas then the PUE is 1.14.

Green datacenter from green.ch company (Switzerland) was designed with energy efficiency and reduction consideration. This project is based on energy-optimized datacenter architecture, latest generation of air conditioners, heat exchangers, and waste heat use in new office building.

The container-sized datacenter designed by Nu@ge aims at keeping the PUE under the value of 1.30, using 2 cooling operating modes:

- Total free cooling when the room temperature is within the server specifications. That range is set by the customer, resulting in a PUE value of 1.16.
- Air recycling with air conditioning when the temperature is out of range, resulting in a PUE value of 1.55.

Thus, the PUE relies strongly on the climate conditions, and customer-defined rules. In the case of Nu@ge's *StarDC* at Marne-La-Vallee (France), weather forecast indicates that 80% of the time, the temperature is below 23° C. The theoretical maximal value for the PUE is then

$$PUE_{\text{Nu@ge}} = 80\% \times 1.16 + 20\% \times 1.55 = 1.24 \tag{2}$$

Among the datacenters in Table 2, it is worth noting that StarDC is the only mobile product. Additionally, it can be produced in series and available to third-party companies in constrast to more efficient but proprietary datacenters.

7 | ENERGY-AWARE MANAGEMENT

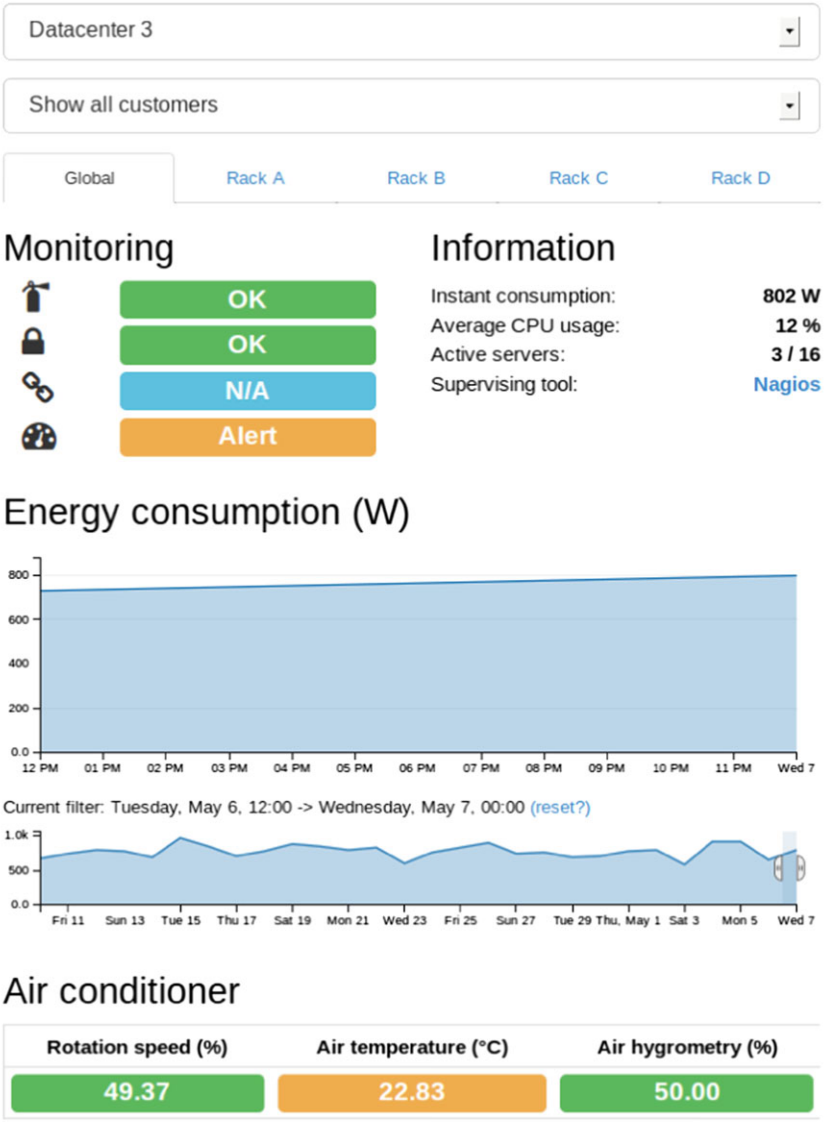
The purpose of the energy-aware management is to evaluate the benefit of green scheduling for reducing electric consumption while matching performance objectives for the virtual machines.

The performance criteria are CPU oriented, and on the basis of a measure of the node performance using all its CPU cores. It produces a value in flops, indicating the number of floating points operations per second. Those benchmarks are based on measurements using ATLAS,[¶] HPL,[‖] and Open MPI.[**] Other criteria exist in the literature, involving the consideration of idle consumption[31] or the use rate[32] of the physical nodes.

[¶] Automatically Tuned Linear Algebra Software.

[‖] Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers.

[**] High Performance Message Passing Library.



Datacenter 3

Show all customers

Monitoring

Information

Instant consumption: 802 W
Average CPU usage: 12 %
Active servers: 3 / 16
Supervising tool: Nagios

Energy consumption (W)

Current filter: Tuesday, May 6, 12:00 -> Wednesday, May 7, 00:00 (reset?)

Air conditioner

Rotation speed (%)	Air temperature (°C)	Air hygrometry (%)
49.37	22.83	50.00

FIGURE 5 Web interface for the visualisation and management of datacenters

TABLE 2 Power usage effectiveness comparison of different datacenters

Datacenter	Company	PUE
Prineville DC	Facebook	1.07
Google DC	Google	1.14
StarDC	Nu@ge	1.24
Green Datacenter	grench.ch	1.4

Abbreviation: PUE, power usage effectiveness. Those values are given by each project but no independent evaluation was done.

Regarding the consumption criteria, 2 approaches are possible. A static way would imply to execute a task on all nodes before starting and measure the power consumption corresponding to the completion time on each node. This method is not significant for long periods because the power consumption of the machine may vary depending on the actual load or external conditions, such as the physical location of the server.

We use a more dynamic approach where the electric consumption metric is based on the number of requests handled by a computationalnode weighted by the power consumption measured during execution. Every time a client submits a request, a computational node will report its electric consumption and total number of requests.

We coupled the scheduling process to resource provisioning while taking into account energy-related events such as fluctuations of electricity prices or heat peaks. In previous work,[25] the present authors proposed methods for provisioning resources and distributing requests with the objective of meeting performance requirements while reducing energy consumption. *GreenPerf*, a hybrid metric, was introduced as a ratio of performance and power consumption for energy efficiency. On the basis of this work, we enable autonomic decisions from the scheduler by checking predefined threshold before executing placement/provisioning decisions.

7.1 | Autonomic and adaptive resource provisioning

We demonstrate the behavior of the scheduler by considering fluctuations of 2 metrics over time, namely, the cost of electricity and temperature. We inject energy-related events at the scheduler level while a client, aware of the number of available nodes, submits a continuous

```
<timestamp value="1385896446">
  <temperature>23.5</temperature>
  <candidates>8</candidates>
  <electricity_cost>0.6</electricity_cost>
</timestamp>
```

FIGURE 6 Sample of the server status

flow of requests intending to reach the capacity of the infrastructure. Requests are scheduled as they arrive to ensure dynamicity.

For the sake of simplicity, the cost of energy is defined as a ratio between the cost over a given period and the theoretical maximum cost. Related to the cost of energy, we defined 3 states:

- Regular time, when the electricity cost is the highest (1.0).
- Off-peak time 1, when the electricity cost is less expensive than during regular time (0.8).
- Off-peak time 2, when the electricity cost is the least expensive (0.5).

Heat measurements are defined through 2 states, depending of the temperature of use: in-range temperature (<25°) and out-of-range temperature (>25°).

The status of the platform corresponds to the value of the exploited metrics at *t* time. The master agent checks the status of the platform every 10 minutes, with the ability to get information about the scheduled events occurring at $t + 20$. Figure 6 presents a sample of provisioning planning, which is a shared XML file using a reader-writer lock that refers to a specific time stamp. For each sample, we defined 3 tags, namely, *temperature*, *candidates*, and *electricity_cost*. At each time interval, the scheduler performs decisions according to the value of the tags. Thus, future information, such as forecasts, can be added to the provisioning planning, ensuring a dynamic behavior regarding to the various contexts. The tags and time interval are customizable variables that can be adjusted to fit specific contexts.

We set thresholds whose values trigger the execution of actions. Actions can be defined through scripts or commands to be called by the scheduler. In this example, we implemented 5 behaviors associated with the experiment metrics. Let *c* be the cost of energy for a given period and *T* the temperature measured at *t*.

- If $T > 25$ then candidate_nodes = 20% of all nodes.
- If $1.0 \geqslant c > 0.8$ then candidate_nodes = 40% of all nodes.
- If $0.8 \geqslant c > 0.5$ then candidate_nodes = 70% of all nodes.
- If $c < 0.5$ then candidate_nodes = 100% of all nodes.

Four different types of events are injected in the provisioning planning made by the scheduler. These events, in turn, fall into 2 categories, namely, scheduled and unexpected. Figure 7 presents how the number of candidate nodes and the energy consumption evolve over time. The left *y*-axis shows the total number of nodes in the infrastructure; the plain line presents the number of candidates during the experiment; the line with crosses is the evolution of the energy consumption, using the right *y*-axis. Each cross describes an average value of energy consumption measured during the previous 10 minutes. The *x*-axis represents the time with a total of 260 minutes.

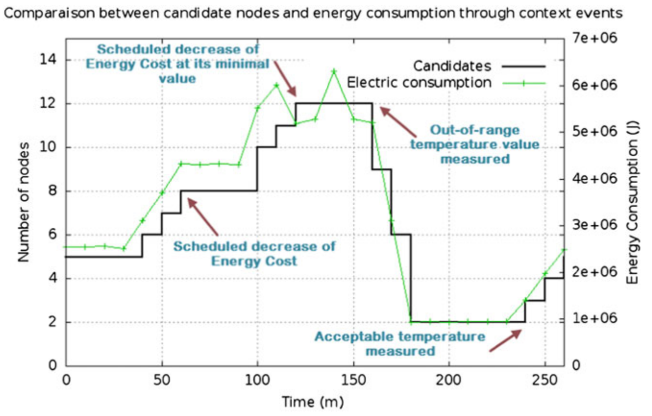


FIGURE 7 Evolution of candidate nodes and power consumption

The infrastructure is deployed on GRID'5000, on the nodes defined in Table 3. The experiment starts with an energy cost of 1.0 and a $Preference_{provider}(u,c)$ giving priority to energy-efficient nodes. The $Preference_{user}$ is not having any influence in the current scenario as the client dynamically adjusts its flow of request to reach the capacity of available nodes.

Event 1 (scheduled) is a decrease of the electricity cost occurring at $t + 60$ minutes. The master agent becomes aware of the information at $t + 40$ minutes. Observing a future cost of 0.8, the agent plans ahead to provide 8 candidates nodes at $t + 60$ minutes. The set of candidates is incremented slowly to obtain a progressive start, at $t + 50$ minutes and $t + 60$ minutes. (It avoids heat peaks due to side effect of simultaneous starts.) We observe a linear increase of electric consumption through the infrastructure. After each request completion, the client is notified of the current amount of candidates nodes and is free to adjust its request rate.

Event 2 (scheduled), similar to **Event 1**; the electricity cost allows the use of every available node in the architecture. The nodes are added to the set of candidates during the following 20 minutes, resulting in a use of all possible nodes between $t + 120$ and $t + 160$ minutes.

Event 3 (unexpected) simulates an instant rise of temperature, detected by the master agent at $t + 160$ minutes. According to administrator rules, the predefined behavior is to reduce the number of candidates nodes to 2. It is performed in 3 steps, to cause a drop of heat and energy consumption. We allow tasks in progress to complete, resulting in a delayed drop of energy consumption. The system keeps on working with 2 candidates until an acceptable temperature is measured at $t + 240$ minutes (**Event 4** [unexpected]). The master agent then starts to provision the pool of candidates every 10 minutes to reach again the value of 12.

TABLE 3 Experimental infrastructure

Cluster	Nodes	CPU
Orion	4	2 × 6cores @2.30Ghz
Sagittaire	4	2 × 1core @2.40Ghz
Taurus	4	2 × 6cores @2.30Ghz

The scenario of this experiment shows the reactivity of the scheduler and its ability to manage energy-related events by adapting dynamically the number of provisioned resources of the physical infrastructure, therefore the power consumption.

8 | CONCLUSION

Despite their maturity, cloud computing technologies still face difficulties concerning their adoption. In particular, the ability to control and build its own premises, along with data sovereignty, is an open issue. In this work, we described the Nu@ge project that aims at providing the means to build a network of modular datacenters with virtualization of IT services. The cloud architecture offers guarantees of control over the underlying infrastructure, knowledge of data location, and control of different QoS.

The software stack presents the ability to aggregate and supervise data from heterogeneous resources to perform federation-wide decisions, by relying on APIs and customization of open-source components. Validation of the project involved a prototype of container-sized datacenters deployed over the French territory. We also focus on the energy-aware provisioning of servers on the basis of energy price and temperature criterias by the means of real experiments on the Grid'5000 platform.

ACKNOWLEDGMENTS

Nu@ge is a research project funded by the FSN (Fund for the Digital Society, BPI France) as part of the *Investissements d'Avenir* program. The authors acknowledge the support of the Nu@ge consortium, composed of the following members:

NON STOP systems secure cloud solutions provider http://www.nss.fr/

CELESTE Internet provider and manufacturer of the StarDC http://www.celeste.fr/

Oodrive online storage specialist http://www.oodrive.fr/

DotRiver virtual desktop and environment provider http://www.dotriver.eu/

Init Sys private network operator http://www.init-sys.com/

New GenerationSR green IT consultant http://www.newgeneration-sr.com/

LIP6 laboratory of UPMC University, Paris. http://www.lip6.fr/

REFERENCES

1. Moore S. Gartner says worldwide cloud infrastructure-as-a-service spending to grow 32.8 percent in 2015. [Online; posted 18-May-2015]; 2015.
2. Rimal BP, Choi E, Lumb I. A taxonomy and survey of cloud computing systems. *Fifth International Joint Conference on INC, IMS and IDC, NCM'09*, Desc:Proceedings of a meeting held 25-27 August 2009, Seoul,Korea, IEEE; 2009:44–51.
3. Prodan R, Ostermann S. A survey and taxonomy of infrastructure as a service and web hosting cloud providers. *2009 10th IEEE/ACM International Conference on Grid Computing (GRID)* , IEEE; 2009:17–25.
4. Satzger B, Hummer W, Inzinger C, Leitner P, Dustdar S. Winds of change: from vendor lock-in to the meta cloud. *IEEE Internet Comput*. 2013;17(1): 69–73.
5. Ferry N, Rossini A, Chauvel F, Morin B, Solberg A. Towards model-driven provisioning, deployment, monitoring, and adaptation of multi-cloud systems. *CLOUD 2013: IEEE 6th International Conference on Cloud Computing*, Santa Clara, CA, USA; 2013:887–894.
6. Rosenblum M. Vmwares virtual platform. *Proceedings of Hot Chips*, vol. 1999; 1999:185–196.http://www.hotchips.org/archives/1990s/hc11/
7. Buyya R, Ranjan R, Calheiros RN. Intercloud: utility-oriented federation of cloud computing environments for scaling of application services. *Algorithms and Architectures for Parallel Processing 10th International Conference, ICA3PP 2010*: Busan, Korea, Springer; 2010:13–31.
8. Cascella RG, Morin C, Harsh P, Jegou Y. Contrail: a reliable and trustworthy cloud platform. *Proceedings of the 1st European Workshop on Dependable Cloud Computing*, EWDCC '12. ACM, New York, NY, USA; 2012:6:1–6:2.
9. Carlini E, Coppola M, Dazzi P, Ricci L, Righetti G. Cloud federations in contrail. *Euro-Par' 11 Proceedings of the 2011 international conference on Parallel Processing*. Springer-Verlag Berlin, Heidelberg; 2012:159–168.
10. Rochwerger B, Breitgand D, Levy E. et al. The reservoir model and architecture for open federated cloud computing. *IBM J Res Dev*. 2009;53(4):4–1.
11. Celesti A, Tusa F, Villari M, Puliafito A. How to enhance cloud architectures to enable cross-federation. *2010 IEEE 3rd International Conference on Cloud Computing (CLOUD)*,IEEE, Miami, Florida, USA; 2010:337–345. http://thecloudcomputing.org/2010/
12. Telecommunication Industry Association. Tia-942 data center standards overview. *White Paper*. 2006.
13. Sacks D. Demystifying storage networking das, san, nas, nas gateways, fibre channel, and iscsi. *IBM Storage Networking*. 2001:3–11. https://www-07.ibm.com/storage/au/pdf/demystifying_storage_networking.pdf
14. Gibson GA, Van Meter R. Network attached storage architecture. *Commun Acm*. 2000;43(11):37–45.
15. Weil SA, Brandt SA, Miller EL, Long DDE, Maltzahn C. Ceph: a scalable, high-performance distributed file system. *Proceedings of the 7th Symposium on Operating Systems Design and Implementation (OSDI)*, Seattle, WA; 2006:307–320. https://www.usenix.org/legacy/events/osdi06/
16. Shvachko K, Kuang H, Radia S, Chansler R. The hadoop distributed file system. *Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, MSST '10. IEEE Computer Society, Washington, DC, USA; 2010:1–10.
17. Morita K. Sheepdog: distributed storage system for qemu/kvm. *LCA 2010 DS&R miniconf*. 2010.
18. Satran J, Meth K, Sapuntzakis C, Chadalapaka M, Zeidner E. Internet small computer systems interface (iscsi); 2004.
19. Hawkins A. Unused servers survey results analysis. *The Green Grid*; White Paper, vol. 28, 2010.
20. Carrera EV, Pinheiro E, Bianchini R. Conserving disk energy in network servers. *Proceedings of the 2003 International Conference on Supercomputing (ICS-03)*, New York; 2003:86–97. ACM Press.
21. Snowdon DC, Ruocco S, Heiser G. Power management and dynamic voltage scaling: myths and facts. *Proceedings of the 2005 Workshop on Power Aware Real-Time Computing*, Jersey City, New Jersey, USA; 2005.
22. Le Sueur E, Heiser G. Dynamic voltage and frequency scaling: the laws of diminishing returns. *HotPower' 10 Proceedings of the 2010 international conference on Power aware computing and systems*,, Vancouver, BC, Canada; 2010:1–8.
23. Beloglazov A, Buyya R. Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints. *IEEE Trans Parallel Distrib Syst*. 2013;24(7): 1366–1379.
24. Feller E, Rilling L, Morin C. Snooze: a scalable and autonomic virtual machine management framework for private clouds. *CCGRID*. Ottawa, Canada: IEEE; 2012:482–489.
25. Balouek-Thomert D, Caron E, Lefevre L. Energy-aware server provisioning by introducing middleware-level dynamic green scheduling.

Workshop HPPAC'15. High-Performance, Power-Aware Computing, Hyderabad, India; May 2015. In conjunction with IPDPS 2015.

26. Roberts LG, Wessler BD. Computer network development to achieve resource sharing. *Proceedings of the May 5-7, 1970, Spring Joint Computer Conference*, AFIPS '70 (Spring). ACM, New York, NY, USA; 1970:543–549.
27. Caron E, Desprez F. DIET: a scalable toolbox to build network enabled servers on the grid. *Int J High Perform Comput Appl*. 2006;20(3): 335–352.
28. Caron E, Toch L, Rouzaud-Cornabas J. Comparison on OpenStack and OpenNebula performance to improve multi-Cloud architecture on cosmological simulation use case. Research Report RR-8421,INRIA; 2013.
29. Barth W. *Nagios: System and Network Monitoring, 2nd Edition New from*: No Starch Press, San Francisco, CA; 2008.
30. Belady C, Rawson A, Pfleuger J, Cader T. Green grid data center power efficiency metrics: PUE and DCIE. Technical Report White Paper 6,The Green Grid; 2008.
31. Diouri MEM, Glück O, Lefèvre L, Mignot J-C. Your cluster is not power homogeneous: take care when designing green schedulers! *IGCC-4th IEEE International Green Computing Conference*, Arlington, VA USA; 2013.
32. Lim S-H, Sharma B, Tak B-C, Das CR. A dynamic energy management scheme for multi-tier data centers. *ISPASS 2011 : IEEE International Symposium on Performance Analysis of Systems and Software*, Austin, TX, USA IEEE Computer Society; 2011:257–266.

How to cite this article: Balouek-Thomert D, Caron E, Gallard P, Lefèvre L. Nu@ge: A container-based cloud computing service federation. *Concurrency Computat: Pract Exper*. 2017;29:e4049. https://doi.org/10.1002/cpe.4049