Hybrid Clouds as Meteorological Simulation Enablers

Alfonso Quarati, Emanuele Danovaro, Antonella Galizia,
Andrea Clematis, Daniele D'Agostino
Institute of Applied Mathematics and Information Technologies
National Research Council of Italy, Genoa, Italy
{quarati, danovaro, galizia, clematis, dagostino}@ge.imati.cnr.it

Antonio Parodi
CIMA Research Foundation, Savona, Italy
antonio.parodi@cimafoundation.org

Abstract The flexible and *pay-as-you-go* computing capabilities offered by cloud infrastructures are very attractive for high-demanding e-Science applications like weather prediction simulators. For their ability to couple the scalability offered by public service provider with the greater control and customization provided by private clouds, Hybrid Clouds seem a particularly appealing solution to support meteorological researchers and weather departments in their every-day activity. Cloud Brokers interfacing customers with cloud providers, may support scientists in the deployment and execution of demanding meteorological simulations, by hiding all the intricacies related to the management of powerful but often complex HPC systems.
The paper presents a brokering algorithm for Hybrid Clouds aimed at the execution of various instances of the weather prediction WRF model subject to different user requirements and computational conditions. A simulation-based analysis documents the performance of different allocation policies at varying workloads and system configuration.

Keywords: Hybrid Clouds, Meteorological Simulations, Brokering Algorithm

1 Introduction

Extreme precipitation and flooding events are among the greatest risks to human life and property, and represent one of the main issues of the 21st century with significant societal and economic implications. To cope with these issues a Meteorological research major objective is to enable the acceleration and the integration of advances in the everyday forecasts thus improving the environment protection. Actually, predicting weather and climate and their impacts is a crucial task both for research groups as well for civil protection departments. Moreover preventing hazards such as floods and landslides needs to address manifold issues that involve not only meteorology scientists but requires a strong connection and collaboration with the ICT community to explore new technological solutions and approaches [1]. In particular, run prediction systems (e.g. WRF, MESO-NH) in a timely and efficient way, both for research and even more for possible operational application, usually requires the use of high performance computing resources that are both costly and not always easily accessible. In particular the Weather Research and Forecasting (WRF) model is a numerical weather prediction and atmospheric simulation system developed to advance the understanding and the prediction of mesoscale weather and accelerate the transfer of research advances into operations. This is "the reference" model for a large community of users, and reflects flexible, state-of-the-art, portable code that is efficient in computing environments ranging from massively-parallel supercomputers to laptops. The computational capacity of the underline system may actually represent a limitation in the every-day-life work of each meteorology scientist [2].

A key factor of cloud computing is represented by its on-demand, pay-per-use approach towards virtualized and distributed ICT solutions – in contrast to the creation and the maintenance of expensive, tightly pre-configured IT infrastructures, necessary to grant analogous services at the same level of business continuity. Hybrid Clouds (HC) integrating internal (private) and external (public) resources, couple the scalability offered by public Clouds with the greater control supplied by private ones. An (hybrid) Cloud Broker (CB) – acting as an intermediary between users and providers of public Cloud services – may support meteorologists in the selection of the most suitable computational platform depending on their simulation objectives, optionally adding the provisioning of dedicated services with higher levels of quality.

Fostered by the research activities of the FP7 EU Distributed Research Infrastructure for Hydro-Meteorology (DRIHM) project (2012-2015) – aimed to enable Hydro-Meteorology research to make a step beyond the state of the art in the modeling of forecasting chain [3] – in the paper we analyze the performance behavior of a brokering algorithm for Hybrid Clouds in adequately responding to the operational constraints raised by different instances of meteorological models, that, as in the case of WRF, may have various kind of users and computational requirements [4]. We analyzed the behavior of different allocation strategies, through simulation, by comparing the results achieved by the brokering algorithm with respect to different system configurations and type of workloads. To this end the study will take into account both the CB than the user (i.e. meteo researcher and professional) perspectives, by considering metrics such as revenue, user satisfaction and system utilization.

Section 2, introduces WRF, its computational issues and their relations with user requirements. Section 3 briefly reports on some related work. Section 4 presents the rationale under the brokering algorithm. Section 5 details the simulation set-up, while Section 6 presents and discusses the results. In Section 7 some conclusions are drawn.

2 The Weather Research and Forecasting Model

The Weather Research and Forecasting (WRF) model is a numerical weather prediction and atmospheric simulation system designed for both research and operational applications [5]. The effort to develop WRF began in the latter part of the 1990's and was a collaborative partnership among the U.S. Institutions, Universities and Laboratories, in particular the National Oceanic and Atmospheric Administration (NOAA), the National Center for Atmospheric Research (NCAR), and more than 150 other organizations and universities in the United States and abroad, [6]. Indeed, it represents a multi-agency effort to build a next-generation mesoscale forecast model and data assimilation system to advance the understanding and prediction of mesoscale weather and accelerate the transfer of research advances into operations. Its spectrum of physics and dynamics options reflects the experience and the input of the broad scientific community. In fact, WRF has grown to a large worldwide community of users (over 20,000 in over 130 countries), and it is now considered a community model contributed by the many research community developers.

WRF is a next-generation forecast model and reflects flexible, state-of-the-art, portable code that is efficient in computing environments ranging from massively parallel supercomputers to laptops. The model serves a wide range of meteorological applications across scales ranging from meters to thousands of kilometers. Applications include real-time Numerical Weather Prediction, tropical cyclone and hurricane research and prediction, regional climate, atmospheric chemistry and air quality, and basic atmospheric research. WRF allows researchers to simulations using real data (observations, analyses) or idealized atmospheric conditions, [7]. The WRF model is frequently used in operational mode in a very large community, widespread from American continents to Europe, through Asia and Israel[1]. A part from U.S., where the model is run for real time forecasts from different department of the National Oceanic and Atmospheric Administration, as the Global Systems Division and the National Sever Storm Laboratory, and other University, the model is actually used also in Mexico and Uruguay. As for Europe, many national and regional forecasts are obtained exploiting WRF, let us cite the LaMMA Consortium in Italy, the Earth Sciences Department of the Barcelona Supercomputing Center in Spain, the Republic Hydro-meteorological Service of Serbia.

2.1 Computational aspects

The WRF model represents the atmosphere as a number of variables of state discretized over regular Cartesian grids. The core is based on an Eulerian solver for the fully compressible nonhydrostatic equations. The model uses terrain-following, hydrostatic-pressure vertical

[1] WRF Real-time Forecasting, http://wrf-model.org/plots/wrfrealtime.php

coordinate with the top of the model being a constant pressure surface. The horizontal grid is the Arakawa-C grid. The time integration scheme in the model uses the third-order Runge-Kutta scheme, and the spatial discretization employs 2nd to 6th order schemes, [7]. Weather prediction codes are by nature I/O (mostly output) intensive, repeatedly writing out a time series of 3D representations of the atmosphere. Among other possibility, the most common standard used for data representation is netCDF. Parallel implementations of the model are distributed, it is possible to support multi-level parallelism: shared-memory (OpenMP), distributed-memory (MPI), and in hybrid parallel (message passing and multi-threaded) computation modes. Few experimental components have been ported on Graphics Processing Unit (GPU) with CUDA, [8] [9]. The parallel version decomposes the WRF domains in the two horizontal dimensions, thus it requires interprocessor communications between neighbors on most supercomputer topologies; each time-step involves 36 halo and 144 nearest-neighbor exchanges (assuming aggregation), [10]. Data decomposition could exploit two-level of memory: distributed memory cranks and (again within each cranks) shared memory tiles. To simulate real cases, WRF needs initial and boundary conditions that may be provided by different global circulation models and reanalysis datasets. Physiographic data (digital elevation model and land use) are also needed.

In the research version, the model supports horizontal nesting that allows resolution to be focused over a region of interest by introducing an additional grid (or grids) into the simulation. The nested grids are rectangular and are aligned with the parent (coarser) grid within which they are nested. Additionally, the nested grids allow any spatial and temporal refinements of the parent grid; compared other models allowing the nesting, the major improvement in the WRF's nesting is the ability to compute nested simulations efficiently on parallel distributed-memory computer systems.

An official benchmark is distributed with the aim of demonstrating computational performance and scalability of the model on target architectures [11]. Given the version of the code, and the test case, performance is expressed through model speed, ignoring I/O and initialization cost, and measured as the average cost per time step over a representative period of model integration. Scalability is given by the ratio of increase in simulation speed to the increase in the number of parallel processes. This is a measure of integration speed, and it would provide a means for comparing the performance of different architectures and for comparison with other models. Results are available for scientific community, and scientists interested in submitting their results obtained on a specific architecture have to follow a procedure. A benchmark exists also to test the GPU deployment of the model [12]. In this case, key computational kernels (within the dynamics and physics of WRF) are analyzed singularly. The aim is to characterize the performance of the kernels in terms of computational intensity, data parallelism, memory etc. thus to improve the effectiveness of the parallel implementations. Other contributions proposed a User-Oriented WRF Benchmarking Collection, [13] and focus on supporting users in setting up model domains for WRF simulations, along with aiding in decisions related to necessary hardware and software resources for various model configurations. Thus, different demanding configurations are evaluated to support users in the estimation of resources needed to integrate over millions of grid points. These represent typical user needs in complicated and demanding multi-nest and multi-shaped configurations used in regional modeling applications. The evaluated metrics are measured in minutes, and include the time spend in doing I/O, a consistent part of the model; thus it is possible to understand the real consumptions of the model when used in complex configurations. A complex configuration corresponds to highly demanding computations, both in terms of memory space and CPU time. For instance, considering the case of a 3-domain configuration[2] composed of about 17 million grid points and 24 GB of input data files. A WRF computation on an Opteron processors connected through an Infiniband[3], required an execution time for each integration step, from about 20 hours when using 96 cores, to 3 hours with 512 cores. The same configuration run on Intel processors connected through a Infiniband[4] requires about 16 hours with 56 cores and 2 hours with 620 cores.

[2] http://weather.arsc.edu/WRFBenchmarking/3dhrlev.html
[3] http://www.arsc.edu/arsc/resources/pacman/index.xml
[4] http://vsc.ac.at/about-vsc/vsc-pool/vsc-1/

2.2 Users Requirements

Working in both operational and research modes, the variety of forecasts and analyses that can be performed by WRF is very large, with model instances requiring from hours to days of computation and a not negligible memory requirements. Moreover, especially when the model is used for research issues, the computational capacity of the underline system may actually represent a limitation. To support such demanding WRF instances it is evident the relevance of ensuring the availability of powerful resources to complete model executions. In fact, it is a quite common situation that WRF demanding runs may be truncated from HPC system administrators when they violate possible limitations set on wall clock time and/or memory quotas for "general" computations. To cope with this situation, WRF enables the periodic (in simulation time) creation of restart files, i.e. checkpoints, to resume the computation from the last restart point printed out, such recovery mechanism result in increases of CPU and memory usage of each model run.

Notwithstanding these common issues, various requirements arise, depending on the role and the needs of the WRF users, which impact on the selection of the most suitable resources to execute the WRF simulations. For example, if used for weather forecasts, WRF execution usually requires high levels of urgency with strict not negotiable deadline. When used for advanced research aims, large computations on multiple nested domains could require long availability assurances to avoid the overhead due to recovery steps, while deadline may be a minor issue.

In the context of a HC devoted to the execution of different kind of workload, in the paper we considered the coexistence of four classes of users, namely Advanced Researchers (AR), Medium Researchers (MR), Weather Forecasters (WF) and Students (ST). Each user has to express a set of parameters useful to configure the setting of her simulation according to her specific requirements and execution options. This configuration setting has to be declared in the *namelist.input* file, also containing other mandatory information, as for example the microphisics and parameterization to be applied, as well as the domain resolution.

Using WRF for scientific research leads to massive simulations both in terms of CPU and memory requirements. In fact in research case, improvements in predictability of extreme events can be obtained considering run at very high resolutions, namely *cloud-permitting* (i.e. 1-5 km grid spacing) and/or *cloud-resolving* (i.e. 1 km grid spacing or less) simulations. For example, an improvement in modeling atmospheric scenarios inducing flash floods could be achieved with their characterization in terms of kinematics, thermodynamics and microphysics properties [10][14]. The use of HPC resources is essential to enable such studies at very high-resolution, involving the exploitation of nested domains. Depending on the degree of accuracy required by the simulation activity we differentiated between medium and advanced researchers. The first require a two-domains simulations at cloud permitting resolutions. The latter require the execution of three-domains simulation, of which the first two domains at cloud permitting resolutions, and the third one at cloud resolving resolution. Due to the computational requirements of researcher's instances, the duration of the runs is actually relevant, also when employing highly parallel resources (see Table 1). In particular, to support AR's heavy computations, without incurring in possible time costly interruptions, it is important that the broker guarantees the highest availability of the scheduled HPC resources. For this reason AR's request of execution are scheduled on the HC private zone. Conversely less demanding MR requests, may be allocated, at a reduced cost (see Table 2), independently on the private or public zones, without have to grant the same level of availability.

When used in operational mode WRF usually run at cloud-permitting resolution, with a high degree of urgency expected by weather forecasters therefore requiring executions on supercomputers or on dedicated clusters. To respect such hard constraint on response time (in the order of hours) the CB must guarantee WFs that their executions are always satisfied in time by supplying powerful parallel resource thus to obtain the fastest results. As explained in next section, to enforce strict deadline requirement, WF requests will always be privileged respect the others.

Students attending a course should learn how to run and configure the model, rather than deeply analyzing simulation results. In this case performance is not the main issue, nor the complexity (i.e. number of grid points and levels of nesting) of the domain. Therefore students' runs of the model do not require high level of parallelism, long computations and large memory space. A student requests may find space indifferently on private or public resources.

For each class of users, Table 1 summarizes the configurations of the WRF instances, along with the simulated period and the execution times as resulted on the cluster detailed in Section 5.3. The second column reports the extension (in km) of the geographical domain to which the simulation applies. The domain is partitioned in a 3D regular grid, according to the spatial resolution determines by columns three and four: first the horizontal resolution (in km), then the number of points in each dimension of the geographical grid[5]. The temporal resolution (number of computed time-steps) is specified in the fifth column. The total duration of the simulated period is given in column six, while column seven reports the computational time required running the proposed configuration on the testbed machine. Finally, last column express specific requirements (if any) of the submitted requests. Each raw of the table specifies each domain involved in the simulation. As previously stated researcher requires complex computations involving up-to three nested domains. From table it is clear how execution time is strictly related to both spatial and temporal resolution as well to the levels on nesting required.

Table 1 Configurations of WRF instances for different class of users

User	Dataset (Km)	Horizontal resolution (Km)	Number of points Nx/Ny/Nz	Timesteps (s)	Simulated period (h)	Execution time (h)	Requirements
MR	1000 400	5 1	200/200/84 400/400/84	2 2	24-72 24-72	22-66	-
AR	1000 400 200	5 1 0.2	200/200/84 400/400/84 100/100/84	1 1 1	12-36 12-36 12-36	37-111	High Availability
ST	500	5	100/100/32	30	24-72	1-4	-
WF	1000	2.5	400/400/40	10	24-72	2-6	Strict deadline

Several commercial [15][16][17] and open source [18][19][20] Cloud broker platforms and tools support organizations in consuming and maintaining cloud services, being IaaS, PaaS or SaaS, particularly when they span multiple providers. These systems generally support the creation and the management of various cloud environments – whether public, private or hybrid clouds – and may be provided on-premises, as SaaS or either solutions. Almost all these tools grant some QoS feature like security, reliability and resiliency. Scientific organizations, like universities and research centres; weather operational departments belonging to public administrations or SME operating in the ICT sector, can advantage of cloud brokers platforms, to exploit their internal ICT infrastructures, combined with public IaaS providers, making the experience gained in configuring and executing specific Meteorological packages, such as WRF, at disposal to third party research groups.

To coordinate the selection of internal and external resources the brokering approach for hybrid Clouds, presented in [21], may be adopted for providing a customizable delivery of meteorological services. In the following, we analyse the economic impact of this brokering methodology respect to a CB and to her, here above described, classes of WRF users.

3 Related Works

Clouds are increasingly being used by businesses, governments and scientists in areas from astronomy to zoology. Many experts affirm that the demands of 21st century science mean that eScience will be largely compute in the Cloud. This vision is based on many reasons, foremost, computational requirements of scientists are bursty, needing massive capabilities for short periods of time, and the cloud supports for short temporary peaks in resource needs. The 21st century science frequently requires the sharing of large datasets and their process through an adequate computational infrastructure, hosting dataset and computing service in the clouds will be much easier and faster than purchase physical devices, with higher flexibility in setting the environment, and replication opportunities. Furthermore, applications are more and more multidisciplinary and progress in science increasingly requires collaborations among many distributed groups. The cloud can facilitate these collaborations. Of course, many barriers in the usage of cloud computing for

[5] Actually, Nx and Ny are the number of points of the horizontal grid, while Nz is the number of vertical levels considered in the configuration.

research and science are still present, those include the perception of "too early", i.e. many open issues have to be studied, and many unresolved issues on how, in particular, domain scientists can best leverage and exploit cloud computing, and hybrid infrastructure composed by cloud and grid resources. [22, 23].

It is possible to find different projects exploiting the clouds to design and/or to build specific eScience environments o to enable the porting of specific codes on the cloud in various application domains. For example in [24], the implementation of a MODIS (Moderate Resolution Imaging Spectroradiometer) satellite data re-projection and reduction pipeline in the Windows Azure cloud computing platform is presented. Authors claim to hide data complexities, processing and transformation from end users, and to dynamically scale to satisfy scientists' various computational requirements in a cost-efficient way. Authors affirm that their experience has further demonstrated the potential of using cloud computing to lower a series of data entry barriers for eScience by leveraging the scalable storage and computing services, and in the same time, outline the unresolved issues related to the use of the cloud and troubleshooting problems defined as "not only time-consuming, but sometimes very tricky". In the field of hazardous chemicals releases, a hybrid cloud computing platform is proposed to solve the problem of hazardous chemicals releases monitoring and forecasting, [25]. Authors present hybrid cloud architecture, called Hazardous Chemicals Monitoring Cloud (HCMC), which provides hybrid management for HPC and Virtual Private Cloud, and storage resources for hazardous chemicals releases data processing and analysis. A scheduling algorithm and QoS policy to assure efficiency of the platform is discussed. Although authors affirm that HCMC is still in a theory-research stage, considering that many chemical domain specialists believe that clouds represent the next generation of mass computing services, future version of HCMC will be provided as a service to process data collected from huge numbers of distributed wireless sensors. In the context of natural language processing (NLP), GATECloud.net has been developed as a cloud-based platform for large-scale NLP research exploiting the Amazon cloud [26]. GATECloud.net is used to layer the existing GATE framework on top of Amazon's cloud infrastructure and services, and to shield users from infrastructural issues as data management, and deployment of virtual machines. The platform has been made available to the public as a beta service (June 2011). During its first six months of operation, the number of processed documents was 4.7 million, amounting for an accumulated server time of 430 h. This level of usage indicates a need for such tools and a clear interest from researchers and the wider community.

Actually the benefits of a possible use of the cloud are appealing also for Hydro-Meteorological scientific community, thus a part from the use of Grid computing [27, 28, 29] it is possible to cite Hydro-Meteorological applications running on the Cloud. For example, the HydroNET portal (www.hydronet.eu) [30] supports water managers in assessing historical, current and forecasted precipitation events, through the combination of different tools already developed. Such combined solution provides a simplification for the work of hydrological practitioners of municipalities and water boards; municipalities use the portal in operational management of excessive rainfall and floods and for post-flood analysis, as a decision support system. HydroNET has been developed as a SaaS cloud application and authors claim use of a hybrid cloud. The private cloud appears to be used as the primary computing environment, managed by a provider of information; while the public cloud for an expansion of computing power when needed. In [31] authors present CloudCast, a mobile application that provides short-term weather forecasts aimed at the real-time nowcasting on a specific geographical location. CloudCast leverages pay-as-you-go cloud platforms, and has two components: 1) an architecture linking weather radars to cloud resources, and 2) a Nowcasting algorithm for generating accurate short-term weather forecasts. This implies a significant data staging to the cloud, which determines possible bottlenecks. To analyze the compute feasibility of cloud services for real-time application of short-term weather forecasting, authors tested four cloud services, two commercial cloud services—Amazon's EC2 and Rackspace Cloud Hosting—as well as two research cloud testbeds—GENICloud and ExoGENI cloud.

As previously mentioned WRF is a highly demanding application usually requiring HPC resources. For this reason distributed infrastructures have been take into consideration for its running. WRF4g is a framework for the execution and monitoring of WRF on Grid environments [32]. The framework claims the ability of running experiments on different computing resources in a transparent way through the DRM4G (Distributed Resource Manager), which allows the user to merge local and remote resources, and Grid infrastructures. Other examples of the WRF porting on the Grid are provided in [33,34]; both works give a detailed description of the strategy used in the development of the Grid implementation, and leverage specific Grid infrastructure as SEEGRID-SCI for the former and the D-Grid for the latter. The former emphasizes the big need and the difficulties occurred for porting the WRF model to the Grid. The latter proposes a performance measurement through the run of the official WRF benchmark [11], and discuss specifically performance issues. An example of WRF running on the cloud is given in [35]. To improve weather forecasting in Central America, the SERVIR team, in collaboration with NASA's Short-term Prediction Research and Transition (SPoRT) Center, is using the NASA Cloud Services to forecast thunderstorms and other weather-related hazards in Central America. SERVIR is producing daily runs of the WRF model to capture large scale weather features moving through the area, as well as details about individual thunderstorm locations and local impacts of terrain on temperature, wind, and precipitation. SERVIR is the US Regional Visualization and Monitoring System with the sponsorship and active participation of NASA and USAID.

4 Brokering algorithm for HC

In [21] we presented a QoS aware brokering algorithm for the allocation of request of execution of business services toward an HC infrastructure. In such composed environment, the computing resources managed by a CB may be supplied by one or more public Cloud provider (e.g. AWS) and by the CB selecting from its internal asset.

In the cited scenario, two classes of service requests were considered: requests with specific QoS requirements (i.e. security, availability) which demanded for a protected execution on the CB internal resources; and requests that do not require any particularly privilege that can therefore be executed on any available resource.

In the following paper we adapt this approach to deal with heavy computation demanding meteorological requests (i.e. various kind of WRF execution), with a further constraint related to the class of Weather Forecasters (WF), that is characterized by the need to be mandatorily (i.e. always) executed in time. We consider therefore the class of demanding requests, to be subdivided into two: WF and AR, with the latter that, still requiring to be executed in-house, can be rejected (i.e. delayed) if no space is available. The other requests (i.e. MR, ST) can indifferently be executed in the private or public zone.

The requested services are delivered by the CB through VMs equipped with the necessary software (i.e. WRF instances with configured parameters according to the related namelist.input file). Based on the brokering algorithm CB allocates the VM to the private or public Cloud zone, depending on the availability of feasible private resources and according to service' computational demand and non-functional constraints. The purpose of the brokering algorithm is of maximizing the number of satisfied users, by reducing the rejected requests, along with the CB's revenue, by maximizing the execution of the VMs on its private Cloud.

4.1 Brokering strategies

According to Table 1 in the following we call, Private cloud Zone ($\mathcal{PZ}$) service, a request of service execution that, due to its specific requirements (e.g. urgency, high availability) need to be executed on a dedicated set of resources (i.e. WF, AR), and Any cloud Zone ($\mathcal{AZ}$) service, a request that can indifferently executed on the private or the public zone (i.e. MR, ST). Allocation strategies aimed at satisfying the majority of $\mathcal{PZ}$ requests, tend to be more conservative hence less-profitable. On the other side, an allocation mechanism that indiscriminately assigns $\mathcal{AZ}$ requests to private nodes increases the number of missed or delayed $\mathcal{PZ}$ requests. To balance between CB eagerness and user satisfaction our allocation solution use part of the private resources to run $\mathcal{AZ}$ services (aimed at increasing the total revenue), reserving a reasonable amount of them free thus to satisfy possible future $\mathcal{PZ}$ requests.

Considering the CB operating a HC composed of two administrative zones, the private and the public one, amounting for N and P (with $N<<P$) physical servers respectively; the brokering algorithms schedules job by applying heuristics based on different quota Q, $Q \leq N$, of private resources dedicated to the execution of type $\mathcal{PZ}$ requests. Depending on the value of Q, the brokering algorithm allows depicting three allocation patterns, namely zone strategies: Feasible ($Q = N$), Static Reservation ($0 < Q < N$) and Max Occupation ($Q = 0$). With Feasible (FE) all private resources are dedicated to perform $\mathcal{PZ}$ requests only (i.e. all $\mathcal{AZ}$ are executed on public zone).

According to Max Occupation (MO), no resource is used exclusively to perform $\mathcal{PZ}$. Static Reservation (SR) reserves a fixed quota of resources Q to execute $\mathcal{PZ}$ requests and lets N-Q resources free to execute the other kind of requests (i.e. $\mathcal{AZ}$). As we discuss in Section 6, the choice of Q affects both CB's revenue and user satisfaction, and strictly relates to the actual system workload.

Moreover, to cope with the strict deadline requirement related to WF requests (see Section 2.2), the algorithm has been modified with respect to the basic version presented in [21]. Independently of the actual strategy deployed, any request $r \neq$ WF is allocated in-house if, and only if, available space is left to execute any WF requests occurring in the time r is executed. If $r \in \mathcal{AZ}$ it is allocated immediately on the public zone, elsewhere (i.e. r=AR) it is rejected. As we discuss this change to the original algorithm impacts on all performance metrics.

4.2 Performance metrics

The performance of the brokering strategies FE, SR and MO at varying system and workload configurations have been measured with respect to three metrics: CB's revenue, user satisfaction and system utilization.

As to user satisfaction we considered the total number of requests accepted in the simulated period. This amounts for the sum of the throughput X_i [36], i.e. number of requests completed by each server i of the Hybrid Cloud:

$$accepted\_req = \sum_i X_i \qquad i=1,\dots,N+P$$

However, as all $\mathcal{AZ}$ requests are always satisfied (either on private or public zone), user satisfaction is better assessed by *refused_req* the number of $\mathcal{PZ}$ requests refused during the observation period. Given M the total monthly arrival rate of submitted requests:

$$refused\_req = M - accepted\_req$$

The revenue of the CB is function of the service price that, in the case of hybrid cloud, may include the cost to rent resources from a public provider. Given a request to execute, for the time t, a service of type k, a customer has to pay the CB the price p_k:

$$p_k = B_k + t * C_k$$

Where the brokering service price B_k is the fee owed to the CB to handle the request, irrespectively if the request is executed in-house or on the public Cloud. The second term is the actual provisioning price, proportional to the execution time t and varying according to the requirements of the specific service and expressed by the hourly cost C_k. The revenue of the CB to execute for the time t, a request j of class k on server i is:

$$revenue_{ij} = B_k + t * C_k, \qquad i \in \textit{private resources}$$

$$revenue_{ij} = B_k \qquad i \in \textit{public resources}$$

The (monthly) total CB's revenue achieved for its brokering activity, accounts for all the X_i requests executed on each server i of the whole hybrid cloud:

$$revenue = \sum_i \sum_j revenue_{ij} \qquad i=1,\dots,N+P;\ j=1,\dots,X_i$$

The utilization U of the private Cloud, under control of the CB, is given by the busy time U_i of each private server i:

$$U = \sum_i U_i \qquad i=1,\dots,N$$

5 Simulation Set-up

In this section we describe the parameters used to perform the experiments, which are the arrival times and the average service times for the classes of requests, the prices we considered and the composition of the private Cloud infrastructure.

5.1 Arrival times and service demand times

The values of the parameters used in the simulation are based on the information collected in many years of Hydro-Meteorological research at the Inter-University Centre for Research in Environmental Monitoring (CIMA Foundation[6]). As regards the arrival and the service demand times we used synthetic workloads generated by statistical functions. The frequency of arrivals of the requests (λ_k) of each class of service $k \in$ {MR, AR, WF, ST}, during a month is modeled with a uniform distribution, not specifying particular daily (e.g. daytime/night-time) or monthly (e.g. weekdays/weekends) time ranges. With respect to more sophisticated solutions [37], this choice is justified by the fact that at least MR and AR jobs require several days of execution times, and to the end of out analysis, it seems not particularly relevant to distinguish the exact moment of their arrival. The service demand times of the classes of requests are uniformly distributed in the specific time-range of their class (e.g. AR requests lay between 37 and 111 hours), as defined in Table 1.

5.2 Definition of prices

Among the public Cloud providers, Amazon EC2 provides suitable solutions for High Performance Computing applications, so we considered its pricing models[7] for their adoption in our simulation. In particular, due to its correspondence in term of technical requirements and economic convenience, we adopt, as reference VM instance, the Amazon “C” instance type corresponding to a two 4-cores Xeon X5570 processors. At present the “C” instance is offered at \$1.3 per hour according to the US East (N.Virginia) region prices in September 2013[8].

According to the definitions of Section 4.1 we assumed that $\mathcal{AZ}$ customers (i.e. MR, ST), not requiring any specific QoS, will pay the Amazon price plus the brokering service B_k that the simulator computed as a 5% of the provisioning price. This solution is reasonable as the brokering algorithm is always able to allocate their requests (at least) on the public Cloud on the basis of the scheduling strategy and the private Cloud load. For what concerns $\mathcal{PZ}$ customers, as WF and AR respectively require urgency and high availability assurance, they are always scheduled on CB’s private zone, therefore major prices are eligible from the CB to execute such requests. Moreover as WF requests have always being satisfied in-time, we assumed for them a price higher than the one paid by AR customers (see Table 2).

5.3 Private Cloud system configuration

To define the configuration of the simulated private Cloud infrastructure we started by considering a series of experiments carried out on the SUPERMUC’s migrations system[9] accessed within the context of the EXtreme PREcipitation and Hydrological climate Scenario Simulations (EXPRESS-Hydro) project[10] that performs high-resolution regional dynamical downscaling of climate scenarios produced by a global climate model, using a state-of-the-art convective model. The performance results achieved resulted in the data reported in Table 1. Based on those results we then performed some benchmarks on the Amazon EC2 instance types and we derived the number of VM instances and the service time for each class of requests as reported in Table 2, on the basis of the “C” instance type. We assumed that the resulting cloud configuration is composed up-to 120 nodes equipped with two 4-cores Xeon X5570 Processors, 23 GB of Ram, linked together via a 10 Gigabit Ethernet network.

Table 2 Simulation parameters

User	Monthly Arrivals	#VM instances	Amazon Instance Type	Cores	Service (h)	$\mathcal{PZ}$ Price/h	$\mathcal{AZ}$ Price/h
MR	8	16	C	8	22-66		\$ 1.3+5%
AR	4	64	C	8	37-111	\$ 2.4	

[6] http://www.cimafoundation.org
[7] http://aws.amazon.com/ec2/pricing/
[8] http://aws.amazon.com/ec2/instance-types/
[9] http://www.lrz.de/services/compute/supermuc/
[10] http://www.lrz.de/projekte/hlrb-projects/0000000000F43551.html

ST	60	1	C	8	1-4		$ 1.3+5%
WF	28	16	C	8	2-6	$3.6	

6 Simulation Results

We have compared the FE, SR and MO zone allocation strategies, by analyzing the achieved CB revenues, users' satisfaction and system utilization at increasing workloads. Due to WF requests peculiarity, i.e. the requirement of always been timely served, we first examined the results achieved when satisfying such constraint, with the ones obtained by applying no reservation on private resources. This comparison sheds light on the algorithm behavior in dealing with more complex situations than the one previously studied [21], where only two categories of request were considered. We then examine metrics variations at doubling system configuration.

In order to ensure meaningfulness of simulated measurements a large number of iterations have been carried out. In particular, for more than 100 iterations, we have not got appreciable differences in the obtained results. The value of parameter Q for the SR strategy has been set to 50% (e.g. half the private resources) for every scenario considered. All simulations take place in a timeframe of one month, and monthly workload (λ) varies between 100 and 400 requests according to the per-class subdivision listed in Table 2.

6.1 Basic scenario

Figure 1 shows the expected monthly average revenue at increasing λ. For all the three policies we can notice revenue rising at λ increases. We also see that this increase tends to proportionally diminish for high values of λ. MO policy allows higher revenues than the other two for all load rates. This fact is due to MO ability of accepting a greater number of $\mathcal{AZ}$ requests respect than SR and FE. The latter, in particular, neglects any $\mathcal{AZ}$ request thus renouncing to cash the provisioning price (i.e. $t * C_k$). Anyway, the gap amongst FE and the other two strategies constantly reduces at higher loads, due to the inability of the private resources to host an increasing number of requests. It is quite surprising that notwithstanding MO has potentially the whole set of private resources at its disposal its major revenue compared to SR is almost negligible. It seems that something hinders its ability to fully exploit such greater asset. Actually this fact is reasonably explained if consider that for satisfying WF requests a lot of resources are left idle, as shown by the utilization graphs of Figure 2.

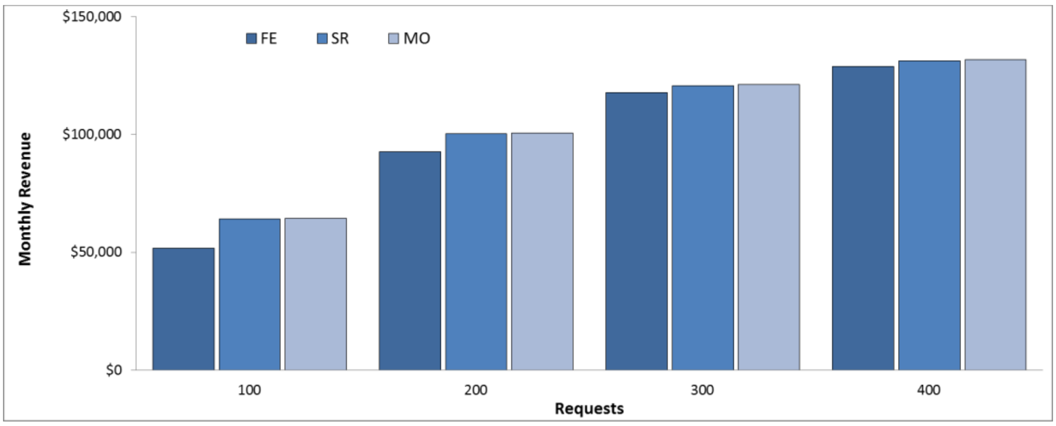

Figure 1 Monthly CB's revenue.

Indeed for all three strategies we observe that utilization is always under 70% even at higher loads. For all loads SR is almost superimposed to MO, justifying their quite identical revenue results. Moreover for $\lambda > 200$ FE curve tends to approximate the others two. Thus at least 30% of resources are actually unavailable for all the strategies just to guarantee the timely execution of WF; if we look at the user satisfaction graphs in Figure 3 we can see how this restriction penalizes AR requests.

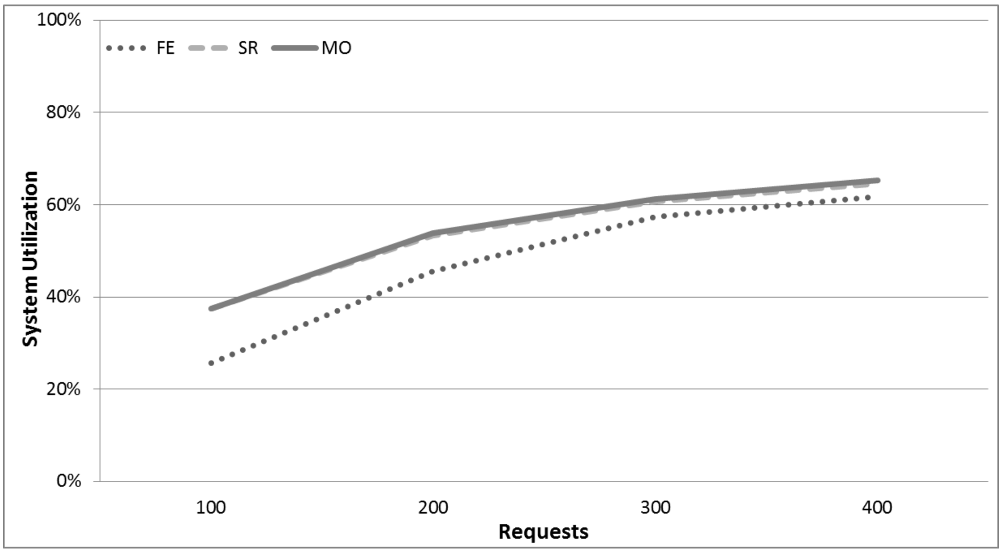

Figure 2 Private zone utilization *U* at varying arrival rates.

Each histogram in Figure 3 depicts, for each λ/strategy pair, the total of AR requests refused. Indeed these are the only possible refused, as WF requests are always mandatorily executed and the $\mathcal{AZ}$ ones can always be address to the public Cloud. With a monthly arrival rates of AR workload assuming values in the set {4, 8, 12, 16} we observe a percentage of refused of 50%, 75%, 91% and 93%. This extremely poor performance is caused by the conservative nature of the brokering algorithm, aimed to fulfill WF requests, along with the heavy computation required by AR and by the (limited) number of system private resources. Indeed a medium AR run employs 512 cores (out of 960) for an average period of 74 hours. Let us call *ar* such request. Even considering the base case (λ = 100), during that period up to four WF requests could arrive (one per day) each requiring 128 cores, plus almost one MR (having a weekly rates of two requests) and several (circa 7.5) ST ones. If the last two classes do not affect FE strategy, we see that in the worst case if four WF should arrive during the time *ar* is running at least one should be rejected. To avoid such situation *ar* is refused. Worst things occur for SR and MO that use part of private resources to run $\mathcal{AZ}$ requests. This attitude is further amplified at heavier workloads.

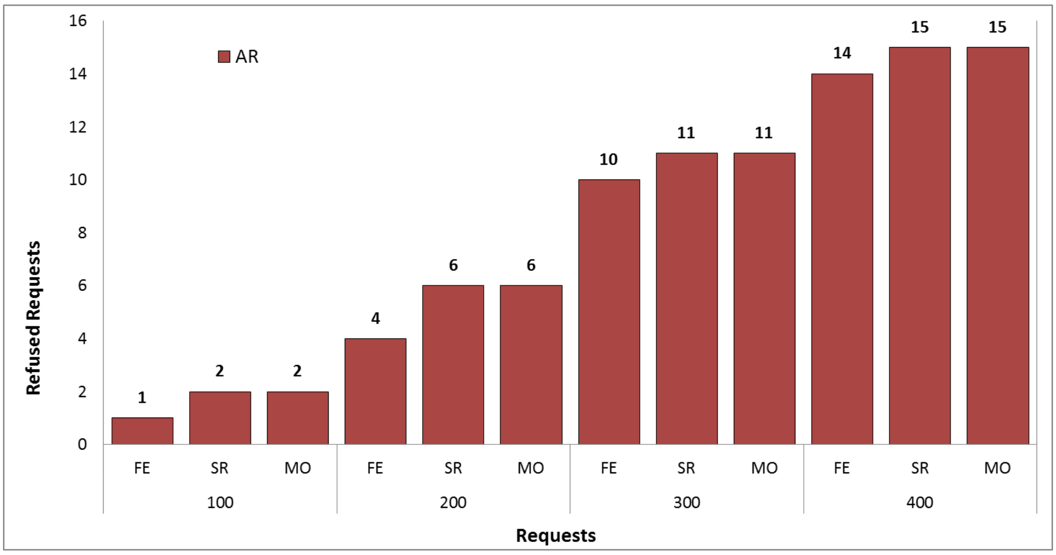

Figure 3 Average monthly number of refused AR requests.

Before considering what happen to the *refused_req* metric if system is upgraded, to ameliorate AR users satisfaction, it is worthwhile to compare the brokering behavior when no limitations is set to always satisfy WF users.

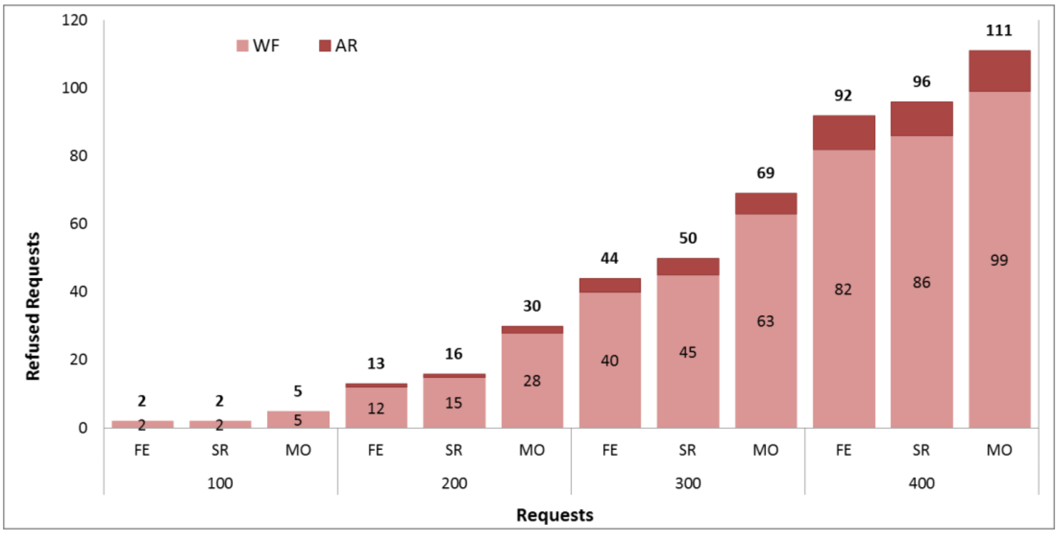

Figure 4 Average monthly number of refused AR and WF requests, when no deadline constraint is applied to WF.

In Figure 4 are reported the number of $\mathcal{PZ}$ requests when no reservation is made to grant WF strict deadline. In this case, as expected, AR users are lesser sacrificed with respect the previous case. When an AR request arrives it is immediately allocated, if there are available resources for its execution, no mind how many space is left for future WF ones. These lasts are indeed greatly penalized in favor of the greedier AR. Such behavior is obviously unacceptable for the situation considered in the present paper subject to the specific requirement of WF. In more general cases, however, as the one presented in [21] the subdivision of user requests in just two categories, if no other distinction is added to $\mathcal{PZ}$ classes, proved to be more suitable to fully exploit the system asset of the HC, as witnessed by the utilization graphs based on that weaker assumption, as reported in Figure 5.

From Figure 5, it is clear the way the three strategies use system resources coherently with their increasing values of Q. Now SR (Q = 50) and MO (Q = 100) curves are neatly separated, showing that the last better exploits the whole set of system resources until saturation. We can conclude that the constraint posed by WF users obliged the CB to not fully exploit its resources; this loss is (partly) compensated by the major price achieved to execute WF requests.

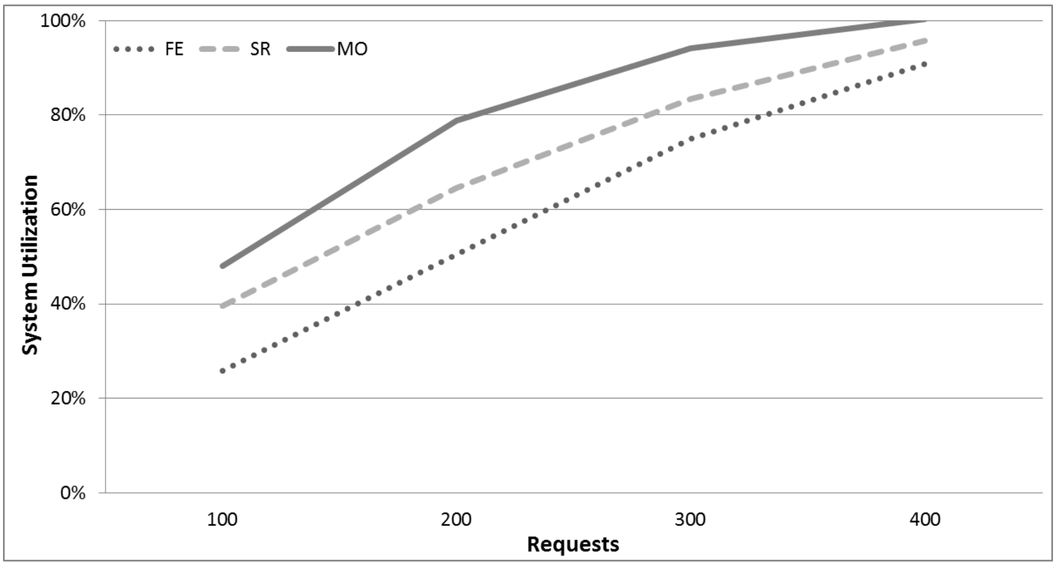

Figure 5 Private zone utilization U at varying arrival rates when no deadline constraint is applied to WF.

Indeed if we look at the CB's revenues when no restriction is set on allocation of $\mathcal{PZ}$ requests, we see from Figure 6 that revenues are still comparably higher than the ones achieved by the basic scenario, notwithstanding the prices for WF have been reduced by a 50% (i.e. 2.4$) and equal to the ones applied for AR. This fact do not surprise if we consider from Figure 4 that, in the general case, several time-consuming AR requests are executed. These requests have an averagely 74 hour execution time against the 4 hour averagely needed to run a WF simulation. Moreover WFs employ 16 VMs against the 64 used by ARs. Thus from the CB point of view this seems to be a preferable solutions, even if WF users suffer a high number of refusals as workload increases.

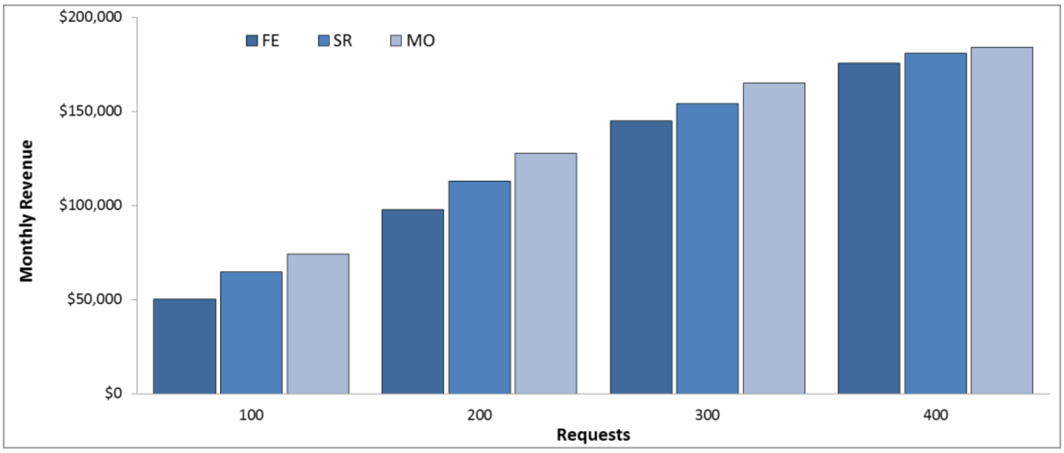

Figure 6 Monthly CB's revenue when no deadline constraint is applied to WF.

6.2 System upgrade

We come back to the initial scenario (i.e. constrained by WF), and analyze the improvements achieved by doubling the number of private resources (i.e. N=60) resulting in a total number of 1920 cores. In Figure 7 the percentage gains achieved by each strategy with respect the basic scenario are reported. With 100% more private nodes available, system is now ready to satisfy a greater number of incoming AR and $\mathcal{AZ}$ requests (see Figure 8). This major system capacity allow to achieve percentage gains up to 95% for MO at λ=400, with good figures for the other load rates and strategies considered. Apart from the lowest rates with just a marginal improvement (1,3%) also FE benefits of the major size of the pool set of nodes up to 62% at λ=400.We can also see from the gap between SR and MO, that the last, due to its ability to allocate all the private resources (Q = 100), is now able to better exploit this pool, thus resulting in noticeable better revenues than SR. Nothing surprising: the more computation availability, the more $\mathcal{AZ}$ requests performed in-house instead turned to the public zone, the higher the revenues. For this reason, both SR and MO always outperform FE, as it completely renounces to the incomes deriving from the execution of $\mathcal{AZ}$ requests.

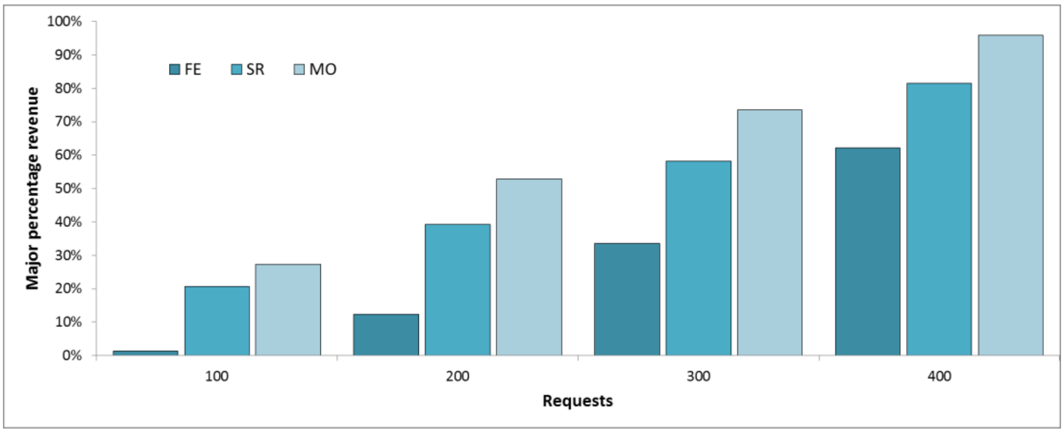

Figure 7 Percentage revenue increases with respect the basic scenario (60 vs. 30 nodes).

The major revenues achieved by the upgraded system firstly result from the ability to satisfy incoming AR requests. Figure 8 outlines the number of AR refusals showing a dramatic improvement, with respect the 30 nodes scenario. For λ<300, no AR is rejected for FE and SR, and just 1 (out of 8) for MO. Even for λ=300 we have 8% of refused (one only) instead of about a 90% of refusals of the basic case, for FE and SR. At this load rate also MO gives an acceptable 16%

percentage of refusals. Things worsen at the maximum workload, even if FE and SR show 12.5% and 25% of AR rejected, almost acceptable if compared to 87.5% and 93.7% of the basic scenario.

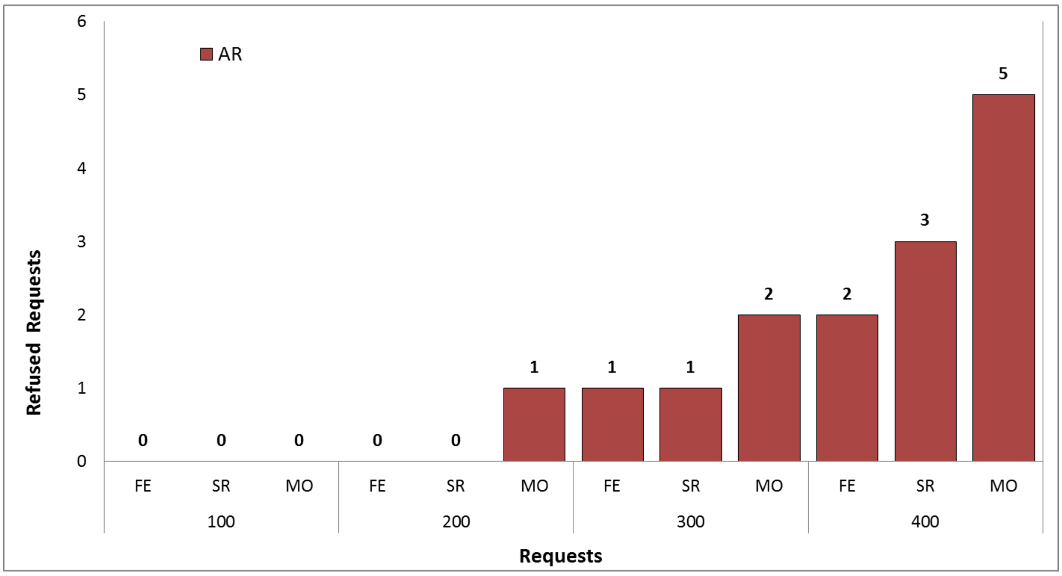

Figure 8 Average monthly number of refused AR requests with a 60 nodes system.

Before concluding, we can summarize that a significant upgrade to the system resources can effectively improve CB's profit as well the level of satisfaction of her customers. The choice of the strategy that better fits to the involved parties' goals, has to be analyzed prospectively by the CB taking into account the trade-off between the two metrics. SR indeed grants a more that satisfactorily level of user satisfactions along with a good revenue response for workload rates up to 300 arrivals at month. If the CB privileges revenues, MO can be a valid option, resulting in the risk to upset a, however, reduced number of AR users (for λ<400). Finally in the case of highest foresee arrivals, FE could be considered the best alternative to satisfy a great number of users (388), penalizing just two of AR ones, but at the expenses to see her revenues significantly be reduced.

7 Conclusions

The *pay-as-you-go* modality offered by Cloud Computing makes it an appealing ICT solution for high-demanding e-Science applications like weather prediction simulators. These complex scientific packages may benefits of a number of Cloud opportunities: the easy deployment of computational (SaaS, PaaS) or infrastructural (IaaS) services; the scalability of the requested hardware that allows to effectively dealing with workload variations; the availability of pre-configured virtual machines that can assist users in quickly performing their simulations. Small-medium research institutions as well public administration departments (e.g. civil protection) operating in the meteorological field, often lack sufficient amount of time and money to create and maintain in-house HPC infrastructures suitable for processing high volumes of weather data at the required speed. Moreover, QoS features such as high availability, resiliency, urgency and enabling factors like customizability are critical points to consider when choosing ICT service providers. All these factors make cloud brokers invaluable actors in the Cloud Computing market for their ability to interface scientific customers with cloud service providers.

In this paper we presented a brokering system for Hybrid Clouds and its adoption to execute various instances of the weather prediction WRF model, by studying the performance of three job-scheduling strategies (i.e. FE, SR and MO). Based on the reservation of a quota of private Cloud resources, the brokering tool manages the allocation of users' requests towards the public or the private Cloud zone depending on the various user requirements, model computational needs and the workload of in-house resources. Simulation results examined the response of the various policies in terms CB revenues, system utilization and user satisfaction.

Differentiating from a previous work, we saw that urgency requirements like the one posed by WF, demand for an over-dimensioned private infrastructure thus to efficiently deals with various classes of users, without too much penalizing some of them (i.e. AR). Once properly dimensioned, the private zone, seems offer adequate space to allocate the great part of requests, at almost the various workloads, and in the meantime is able to grant significant (with respect the initial scenario) revenues to the CB.

From the results analysis followed that, although MO is certainly the best policy as regards CB revenue, it majorly penalizes AR customers, at medium-high load rates. By contrast FE is more customers safeguarding at the cost of diminish CB revenues. For this reasons SR, which follow a less greedy approach, without too much disregards users' needs, seems a good trade-off compromise between CB economic goals and users' satisfaction.

Acknowledgments

This work is partially supported by DRIHM EU FP7-Infrastructure Project under grant, Contract No: RI-28356.

References

1. J. Shukla, T. N. Palmer, R. Hagedorn, B. Hoskins, J. Kinter, J. Marotzke, M. Miller, J. Slingo, Toward a new generation of world climate research and computing facilities. Bulletin of the American Meteorological Society, 91, 1407–1412, 2010.
2. J. Michalakes, J. Hacker, R. Loft, M. McCracken, A. Snavely, N. Wright, T. Spelce, B. Gorda, R. Walkup, WRF Nature Run, , Journal of Physics: Conference Series SciDAC, n. 125, pp.1-6, 2008.
3. A. Clematis, D. D'Agostino, E. Danovaro, A. Galizia, A. Quarati, A. Parodi, N. Rebora, T. Bedrina, D. Kranzlmüller, M. Schiffers, B. Jagers, Q. Harpham, P.H. Cros, DRIHM: Distributed Research Infrastructure for Hydro-Meteorology. In: 7th International Conference on System of Systems Engineering (SOSE2012) Proceedings, pp. 149-154, IEEE Computer Society, 2012.
4. S. Marru, D. Gannon, S. Nadella, P. Beckman, D. B. Weber, K. A. Brewster, K. K. Droegemeier, LEAD Cyberinfrastructure to Track Real-Time Storms Using SPRUCE Urgent Computing. CTWatch Quarterly, Volume 4, Number 1, 2008.
5. The Weather Research & Forecasting Model, http://wrf-model.org/index.php
6. J. Michalakes, J. Dudhia, D. Gill, J. Klemp and W. Skamarock: Design of a next-generation regional weather research and forecast model : Towards Teracomputing, World Scientific, River Edge, New Jersey, pp. 117-124, 1998.
7. W. C. Skamarock, J. B. Klemp, J. Dudhia, D. O. Gill, D. M. Barker, M. G. Duda, X.Y. Huang, W. Wang, J. G. Powers, A Description of the Advanced Research WRF Version 3, NCAR/TN–475+STR, NCAR TECHNICAL NOTE, June 2008.
8. J. Michalakes and M. Vachharajani, GPU Acceleration of Numerical Weather Prediction. Parallel Processing Letters, Vol. 18 No. 4, pp. 531—548, World Scientific, 2008.
9. V. Simek, R. Dvorak, F. Zboril, J. Kunovsky, Towards Accelerated Computation of Atmospheric Equations using CUDA, IEEE Proceedings of the International Conference on Computer Modelling and Simulation, pp. 449-454, 2009.
10. J. Michalakes, J. Hacker, R. Loft, M. McCracken, A. Snavely, N. Wright, T. Spelce, B. Gorda, R. Walkup, *WRF Nature Run*, SciDAC 2008, Journal of Physics: Conference Series 125, 2008.
11. WRF V3 Parallel Benchmark Page, http://www.mmm.ucar.edu/wrf/WG2/bench/
12. GPU Acceleration of NWP: Benchmark Kernels Web Page: http://www.mmm.ucar.edu/wrf/WG2/GPU/
13. User-Oriented WRF Benchmarking Collection, http://weather.arsc.edu/WRFBenchmarking/index.html
14. E. Fiori, A. Parodi, F. Siccardi, Turbulence closure parameterization and grid spacing effects in simulated supercell storms. Journal of the Atmospheric Sciences, 67(12), 3870-3890, 2010.
15. http://www.zimory.com
16. https://www.computenext.com
17. http://www.gravitant.com
18. http://www.compatibleone.org
19. http://stratuslab.eu
20. http://www.reservoir-fp7.eu
21. D. D'Agostino, A. Galizia, A. Clematis, M. Mangini I. Porro, A. Quarati, "A QoS-aware broker for hybrid clouds", Computing, vol. 95, issue 1, pp 89-109, 2013.
22. J. Hellerstein, The use of the cloud for e-Science, in: IEEE Proceedings of the 8th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS '13), pp. 1, 2013.
23. e-IRG Task Force on Cloud Computing, Cloud Computing for research and science: a holistic overview, policy, and recommendations, e-IRG Techincal Report, 2012. Available at: http://www.e-irg.eu/images/stories/dissemination/e-irg_cloud_computing_paper_v.final.pdf

24. J. Li, D. Agarwal, M. Humphrey, C. van Ingen, K. Jackson, and Y. Ryu. eScience in the Cloud: A MODIS Satellite Data Reprojection and Reduction Pipeline in the Windows Azure Platform. In Proceedings of the 24th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2010), Apr 19-23, Atlanta, Georgia, 2010.
25. X Shi, Y Sui, Y Zhao, Hybrid Cloud Computing Platform for Hazardous Chemicals Releases Monitoring and Forecasting, Journal of Computers, Vol. 7, no. 9, pp. 2306-2311, 2012
26. V. Tablan, I. Roberts, H. Cunningham, H. and K. Bontcheva, GATECloud. net: a platform for large-scale, open-source text processing on the cloud. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 371, no. 1983, 2013
27. A.S. Cofino, D. San-Martın, and J.M. Gutierrez, A Web Portal for Regional Projection of Weather Forecast Using GRID Middleware, In Proceeding of International Conference on Computational Science 2007, Part III, LNCS 4489, pp. 82–89, 2007.
28. K. A. Brewster, D. B. Weber, S. Marru, K. W. Thomas, D. Gannon, K. Droegemeier, J. Alameda, and S. J. Weiss, On-Demand Severe Weather Forecasts Using TeraGrid via the LEAD Portal, Proceedings of the Third Annual TeraGrid Conference, 2008
29. O. Terzo, L. Mossucca, A. Albanese, R. Vigna, N. P. Premachandra, A distributed environment approach for a worldwide rainfall hydrologic analysis, In IEEE Proceeding of the International Conference on Complex, Intelligent, and Software Intensive Systems, pp. 271-276, 2011.
30. T Einfalt, A Lobbrechtm, I. Poortinga, Decision support for urban drainage using radar data of HydroNET-SCOUT. In Proceeding of the International Symposium on Weather Radar and Hydrology, IAHS, 2011.
31. D. K. Krishnappa, D. Irwin, E. Lyons, and M. Zink, CloudCast: Cloud computing for short-term mobile weather forecasts. In IEEE proceeding of the 31st International Performance Computing and Communications Conference (IPCCC), pp. 61-70, 2012.
32. V. Fernandez-Quiruelas, J. Fernandez Fernandez, A. S. Cofino, and L. Fita, WRF4G: The Weather Research Forecasting model workflow for the GRID, In EGU General Assembly Conference Abstracts, vol. 12, p. 12973, 2010.
33. D. Davidović, K. Skala, D. Belušić, and MT Prtenjak, Grid implementation of the weather research and forecasting model, Earth Science Informatics, vol 3, no. 4, pp. 199-208, 2010.
34. J. Ploski, G. Scherp, T.I. Petroliagis, O. Buchner and W. Hasselbring, Grid-based deployment and performance measurement of the Weather Research & Forecasting model. Future Generation Computer Systems, Vol. 25, no. 3, pp. 346-350, 2009.
35. https://servirglobal.net/Global/Articles/tabid/86/Article/1158/servir-to-use-cloud-computing-to-improve-weather-forecasting-in-central-america.aspx
36. D.A. Menascé, V.A. Almeida, and L.W. Dowdy, Performance by Design: Computer Capacity Planning by Example, Prentice/Hall, 2004.
37. Q. Nou, S. Kounev, and J. Torres, “Building online performance models of grid middleware with fine-grained load-balancing: a Globus toolkit case study”, *Formal Methods and Stochastic Models for Performance Evaluation, Lecture Notes in Computer Science*, vol. 4748, pp. 125-140, 2007.