Int J Parallel Prog (2017) 45:1164–1193
DOI 10.1007/s10766-016-0457-y

# An Approach to Forecast Queue Time in Adaptive Scheduling: How to Mediate System Efficiency and Users Satisfaction

**G. B. Barone[1] · V. Boccia[2] · D. Bottalico[1] · R. Campagna[1] · L. Carracciuolo[3] · G. Laccetti[1] · M. Lapegna[1]**

Received: 29 February 2016 / Accepted: 22 September 2016 / Published online: 8 October 2016

**Abstract** The minimisation of the total cost of ownership is hard to be faced by the owners of large scale computing systems, without affecting negatively the quality of service for the users. Modern datacenters, often included in distributed environments, appear to be "elastic", i.e., they are able to shrink or enlarge the number of local physical or virtual resources, also by recruiting them from private/public clouds. This increases the degree of dynamicity, making the infrastructure management more and more complex. Here, we report some advances in the realisation of an *adaptive scheduling controller* (ASC) which, by interacting with the datacenter resource manager, allows an effective and an efficient usage of resources. In particular, we focus on

✉ L. Carracciuolo
luisa.carracciuolo@cnr.it

G. B. Barone
giovannibattista.barone@unina.it

V. Boccia
vania.boccia@na.infn.it

D. Bottalico
davide.bottalico@unina.it

R. Campagna
rosanna.campagna@unina.it

G. Laccetti
giuliano.laccetti@unina.it

M. Lapegna
marco.lapegna@unina.it

[1] University of Naples Federico II, Naples, Italy

[2] Italian National Institute of Nuclear Physics, Naples, Italy

[3] Italian National Research Council, Naples, Italy

the mathematical formalisation of the ASC's kernel that allows to dynamically configure, in a suitable way, the datacenter resources manager. The described formalisation is based on a probabilistic approach that, starting from both a hystorical resources usage and on the actual users request of the datacenter resources, identifies a suitable probability distribution for queue time with the aim to perform a short term forecasting. The case study is the SCoPE datacenter at the University of Naples Federico II.

**Keywords** Adaptive scheduling · Resources management · Large scale and distributed systems · Queue time forecasting

## 1 Introduction

The TCO of large scale computing systems, supporting a wide range of users and different applications, includes also the initial hardware cost (for computing nodes, storage systems, racks, facilities, etc.), the personnel/system administrator costs (salaries for software and hardware maintenance requiring specialised know-how), business premises, and energy costs (e.g. the additional power requirements for cooling and power delivery inefficiencies) [15].

Users of such large scale systems, which are often integrated in complex distributed computing environments (e.g. grid and cloud), can have conflicting demands in terms of resources request since they ask to execute very different jobs (i.e., short versus long jobs, sequential versus parallel applications, etc.).

The aim of a good system manager is the TCO minimisation without neglecting all users satisfaction. In other words, the system manager has to "configure" the system in such a way to not "waste" resources (if not required by any user) and to recruit the "suitable" number of resources in order to execute, in a "reasonable time", all the applications submitted by the users.

The overall time to solution of an application on such computing resources is made of a part related to its execution time plus a part spent in waiting that such resources are available on the system. While the execution time of an application on a certain fixed system configuration is a priori known on the contrary, on a large general purpose system, the waiting time depends both on the application requirements (in terms of hardware resources demand) and on the system state at the time of application submission. The user satisfaction depends on the total time required to complete its submitted jobs. Thus, the overall users satisfaction depends on how the system is able to complete the whole jobs work flow.

Modern computing techniques and paradigms (e.g. virtualisation and cloud computing) can help the system manager to suitable activate on demand the right number of the system resources, in order to balance the TCO minimisation and all users satisfaction maximisation.

In the described scenario, an adaptive approach to the scheduling problem seems to be suitable to solve the above optimisation problem.

We are working since some years (see [2,3]) to realise an adaptive scheduling controller (ASC), a complex system that, using an adaptive approach to the scheduling problem, is able to periodically recognise a change in the job work flow scenario in order to verify if the actual system configuration (in terms of scheduler parameters) allows to grant the same level of performances measured by means of classical efficiency and effectiveness metrics. In case of performance degradation, the scheduler has to assume a new configuration increasing the level of performances.

Already in [2,3], the SCoPE datacenter of the University of Naples Federico II has been used as test case to study the issues related to the design and the implementation of ASC. The SCoPE datacenter, in fact, is a distributed computing infrastructure, which has the twofold role of local computing resources provider for the academic research groups and of remote resources provider for both the IGI [9] and the EGI Federated Cloud [6].

This paper describes the advancements in the mathematical formalisation and implementation of ASC's kernel module. Main results concern the definition of a heuristic approach to find the most suitable probability distribution function to model the queue times meanwhile forecasting them for different classes of jobs. The classification is made on the basis of job duration and degree of parallelism (DoP).

In Sect. 2, we provide some details on the related works both in the field of adaptive scheduling and in the field of queue time forecasting, focusing on the key differences between the existing approaches and ours. In Sect. 3, we report a short description of the ASC system, focusing on the mathematical formalisation of one of its important core module. In Sect. 4, we describe the case study of the SCoPE computing infrastructure at University of Naples Federico II. In Sect. 5, the approach used both to characterise queue time and to forecast them by means of a hybrid approach based on numerical and statistical techniques is presented.

## 2 Related Work

According to [5], an adaptive solution to the scheduling problem is able to change dynamically algorithms and parameters defining the scheduling policy, taking into account the *past and present* behaviour of the system and the previous decisions of the scheduling system itself (see [18]).

A preliminary approach in jobs scheduling, such as those described in [19], models adaptive control systems that are able to maximise a given performance criterion, such as system throughput. However, in the last years the heterogeneity of applications using general purpose computing systems grows together with the complexity of resource requirements. Moreover, the overall set of resources, generally included in a modern "elastic" datacenter, is dynamic and heterogeneous itself, because of the chance to include different resources (e.g. from clouds).

In this scenario, characterised by a high level of dynamicity, the system throughput maximisation shouldn't longer be the only requirement for a scheduling scheme. The user perceives a good quality of service (the so called "user satisfaction") if his problem is solved in an *"acceptable"* time [7] and this regards not only the application execution time but all the time from job submission until job completion.

Recent approaches in jobs scheduling take into account both *efficiency* and *fairness* for homogeneous workloads [21], but the open challenge is to achieve the same goal for *not* homogeneous workloads also due to the high computational complexity of the optimisation problem to be solved.

With the aim to reduce the computational complexity for the optimisation problem, in [2,3], we proposed a pragmatic approach, based on the assumption that the overall set of users is organised in communities each of them having almost homogeneous requirements. These conditions induce a classification of the users (or equivalently of the jobs) driven by two parameters: the job duration and the job DoP.

Moreover, the most of the approaches existing in literature, present and discuss the task distribution problem onto HPC, distributed systems or service infrastructures with the aim to improve mainly the performance and load balancing of the applications [1,4,8,11,13,16]. Our approach, instead, aims to achieve the efficiency of the entire system rather than the performance of a single application.

All said is possible only if we are able to find the relationship between a configuration of the scheduling system and the values of the metrics to be optimised.

In our previous work [2], we reformulated the problem of metrics optimisation in such a way to depend only on a queue time estimation. Thus, if we are able to obtain, for each system configuration and each work flow, a queue time estimation, we are also able to forecast values for metrics and, then, to assess whether the current system configuration is good or needs to be changed for a certain work flow.

Here we introduce a probability distribution function (PDF) useful to forecast the queue time. In existing literature many articles have been produced in the field of queue time theory. In [17,22] are well described Queuing Theory and some mathematical models of queuing systems of the last 50 years.

However, in theoretical studies with the aim to reduce problem complexity, these parameters are often simplified up to the point of no longer being representative of a real system. In distributed and on demand systems, e.g., there may be more than a server (thousand of nodes, that may change both in number and in type) and the scheduling algorithm can be not simple but a combination of basic modules.

While the theoretical studies offer refined models, instead the heuristic approaches provide a more realistic system representation. In the last decades, some authors began to use probabilistic approaches to describe queue time in queue systems starting from the dataset related to the work flow data. In some of these works, the dataset used in the probabilistic approach is not real but "simulated" and sometimes related to only one class (e.g., see [21]) of "users" (parallel, sequential, etc.).

In our approach, even if we follow a probabilistic approach in queue time characterisation, however we also consider real data collected during the past year on the SCoPE datacenter (see Sect. 4 for the description of the case study). In this case, the probability distribution for queue time varies in time due to the heterogeneity and dynamicity of jobs in work flow.

According to some recent results [14], we focused mainly on two promising probability distribution function (PDF): the Gamma function and the General Pareto function. There, queue time is represented as the sum of two random variables: the first one having a distribution of Gamma type, the second one of Pareto type.

Our approach doesn't take into account the mathematical formalisation of the PDF sum, but starts from the following assumption: on the basis of the system status, the values for the random variable of the queue time are generated according to a probability distribution that is more akin to a type Gamma or Pareto.

## 3 The ASC Core Module: A *Statistics Optimiser* Formulation

In a dynamic context, where the number of jobs (and their type) may change as well as the number of system resources, an adaptive scheduler has to verify periodically the status of the job work flow and, in case of some changes, it has to verify if the actual system configuration (in terms of scheduler parameters) is the right to grant the same performances measured by means of classical efficiency and effectiveness metrics. If a change in the configuration parameters is needed, an adaptive scheduler controller has to look for a new "optimal" configuration.

We define "adaptive" a system able to "reconfigure" itself on the basis of changing in the user typology. Such a mechanism, analysing system behaviour by some classical *key-statistics* (e.g. depending on queue waiting time, jobs throughput, resource usage, and so on), dynamically defines a new set of scheduler *key-parameters* values. The scheduler's new configuration has to meet both the user satisfaction and the efficiency/productivity in the computational resource usage.

Here we describe the mathematical formalisation at the basis of the ASC's kernel module: the *Statistics Optimiser* (see [2] for a complete description of the ASC architecture and operating model). As in our previous works [2,3], we suppose the whole work flow of heterogeneous jobs partitioned into $m$ classes of homogeneous jobs. Then, $\forall j \in \{1, \ldots, m\}$, we look for a function $F_j$ such that:

$$\mathbf{C} = \sum_{j=1}^{m} \alpha_j(\mathbf{J}) \cdot F_j(\mathbf{S}) \tag{1}$$

where $\mathbf{C}$ is the set of the resource manager configurations (each of them is identified by a set of value for *key-parameters*), $\mathbf{J}$ is the vector representing the work flow, $\mathbf{S}$ contains the values of the considered metrics.

The function $F_j$ computes *optimal* parameters values for the $j$th job type, while $\alpha_j$ expresses the *weight* to be considered for the $j$th job type. In other words, $F_j$ has to act in the following way:

$$F_j(\mathbf{S}) = \begin{cases} \left(c_{i_1, i_2, \ldots, k}\right)_j^{opt} & \text{if } S_k \nsim (S_k)_j^{opt} \\ \text{leaves unchanged } \left(c_{i_1, i_2, \ldots, k}\right)_j & \text{otherwise} \end{cases}$$

For each job class, we consider the following classic *key-statistics*:

*System effectiveness ratio* $E^{(j)} = \frac{\sum_{i=1}^{n^{(j)}} p_i^{(j)} t_i^{(j)}}{P^{(j)} T^{(j)}}$;
*System Make span* $M_k^{(j)} = \max_{i=1,\ldots,n^{(j)}} \left(t_i^{(j)} + q_i^{(j)}\right)$;
*Queue waiting time average* $Q^{(j)} = \frac{\sum_{i=1}^{n^{(j)}} q_i^{(j)}}{n^{(j)}}$

where $P^{(j)}$ is the total number of available processors allocated by job type $j$, $n^{(j)}$ is the total number of jobs in the $j$th jobs class, $T^{(j)} = \sum_{i=1}^{n^{(j)}} t_i^{(j)}$ is the wall clock run time for all jobs in the $j$th class, $p_i^{(j)}$, $t_i^{(j)}$ and $q_i^{(j)}$ are respectively the number of requested processors, the execution time and the queue time for the $i$th job in the $j$th jobs class.

As showed in our previous work [2], and under the *realistic* assumption:

$$\sum_{i=1}^{n} q_i < \sum_{i=1}^{n} t_i \tag{2}$$

for each job class $j \in \{1, \dots, m\}$, we want to solve the following:

**Problem 1** *To compute the set of the scheduler* key-parameters $\mathbf{C}_{Opt}^{(j)}$ *such that*

$$\mathbf{C}_{Opt}^{(j)} = F_j \left( \mathbf{S}_{Opt}^{(j)} \right) \tag{3}$$

*where* $\mathbf{S}_{Opt}^{(j)} = \left( E_{Opt}^{(j)}, {M_k}_{Opt}^{(j)}, Q_{Opt}^{(j)} \right)$ *and* $E_{Opt}^{(j)}, {M_k}_{Opt}^{(j)}, Q_{Opt}^{(j)}$ *are the solutions of the constrained "optimisation" problem:*

$$\begin{cases} \max\{E^{(j)}\} \quad s.t. \\ T^{(j)} \le ({M_k}^{(j)} + n^{(j)} Q^{(j)}) \end{cases} \tag{4}$$

□

Among the described three *key-statistics*, we want to maximise the efficiency and, than, the optimisation problem is defined thanks to the constraints related to users satisfaction (Makespan and Queue Waiting Time average).

We can estimate $E$, $Q$ and $M_k$ only if a PDF for the queue time values is known for all the jobs in the work flow. Once an estimation for the metrics is done, the system can evaluate if the actual configuration is good or has to be changed. In this last case, a new system configuration has to be defined optimising metrics estimation.

In the next sections we describe the approach followed to identify the PDF that allows both to describe and to forecast queue time for all the classified jobs.

## 4 Our Case Study Description

We use computational resources available at the University of Naples Federico II, acquired in the context of PON Italian National Project titled S.Co.P.E. *Sistema Cooperativo Per Elaborazioni scientifiche multidisciplinari* [12]. SCoPE resources are also available both to national and to international relevant distributed infrastructures (IGI and EGI). A large number of applications run on this infrastructure; among them, those ones who more intensively use the system, belong to a some different scientific fields (from Biology to Physics, from Engineering to Numerical Analysis).

**Fig. 1** Jobs *“ideal classification”*

| | | Job duration | | |
|---|---|---|---|---|
| | | *short* | *medium* | *long* |
| DoP | *low* | T0P0 | T1P0 | T2P0 |
| | *medium* | T0P1 | T1P1 | T2P1 |
| | *high* | T0P2 | T1P2 | T2P2 |

Due to the heterogeneity of the user communities, the computational resources are used both for *“traditional”* GRID jobs and for HPC applications. From our heuristic analysis, we observe that SCoPE jobs are mostly sequential or with a low DoP with a short/medium duration. Just a subset of SCoPE jobs has a medium-high DoP and a more long duration.

The computational resources (about 2000 cores) are accessed by means of a Resource Management System (based on Maui-Torque systems).

Jobs are classified on the basis of an *“ideal classification”* driven by the job duration and the tasks number (see Fig. 1). Figure 2 shows the real work flow characterisation during the last year 2015. Different type of work flows and jobs are present on SCoPE infrastructure confirming the need for an adaptive approach to the scheduling problem.

We remark that, with the terms *short*, *medium* and *long* for the job duration, we intend: *short* till to 2 h, *medium* from 2 h to 2 days and *long* above 2 days. With the terms *low*, *medium* and *high* for the job DoP we respectively intend: sequential jobs, parallel jobs with up to eight concurrent tasks and parallel jobs with more eight concurrent tasks.

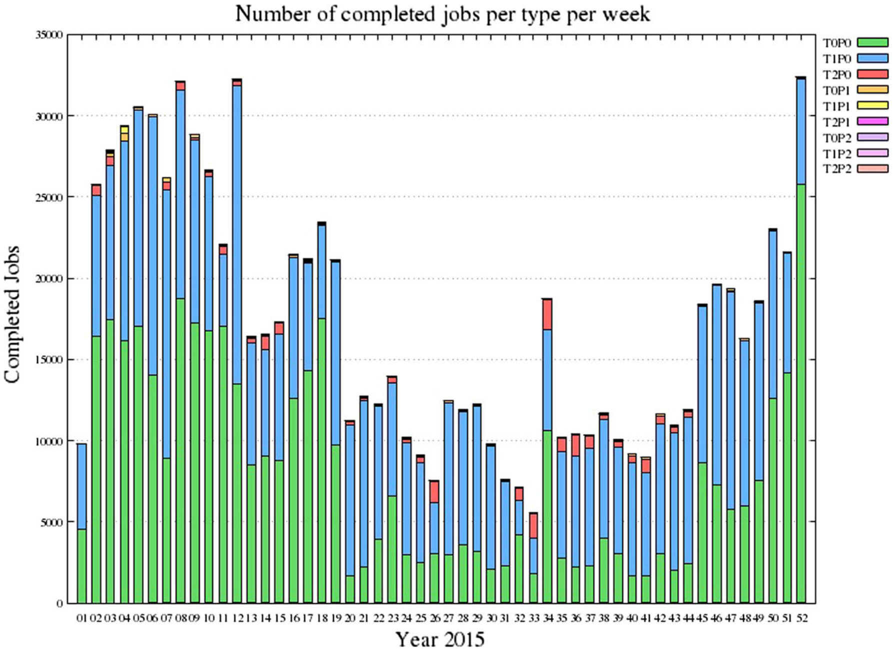


**Fig. 2** Job work flow representation (Year 2015): # of jobs per type versus year week number

# 5 Queue Time Characterisation and Forecast

A mathematical characterisation of the waiting time $q$ is necessary to evaluate some of the metrics defining the optimisation problem (4) (e.g., $M_k^{(j)}$ and $Q^{(j)}$) and it is also useful during the decision process for the definition of the scheduler configuration parameters. One of the approaches used for the characterisation of the queue waiting time provides the chance to define $q$ as a random variable with an associated PDF.

## 5.1 Probabilistic Characterisation of the Queue Time

As described in [14], $q$ can be considered as the sum of two independent random variables $q = q_{notwaiting} + q_{waiting}$ where $q_{notwaiting}$ and $q_{waiting}$ represent the non-waiting and waiting time, respectively: $q_{notwaiting}$ and $q_{waiting}$ are associated to a

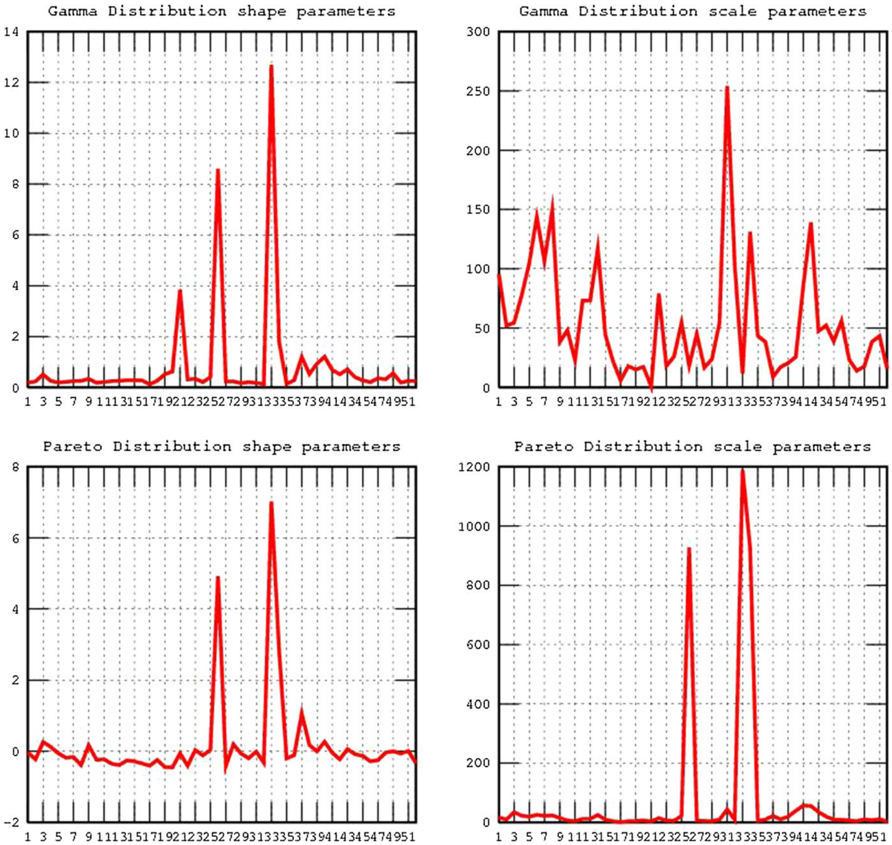


**Fig. 3** Trends of the estimated Pareto and Gamma distributions parameters for medium long sequential jobs for every week of the year 2015

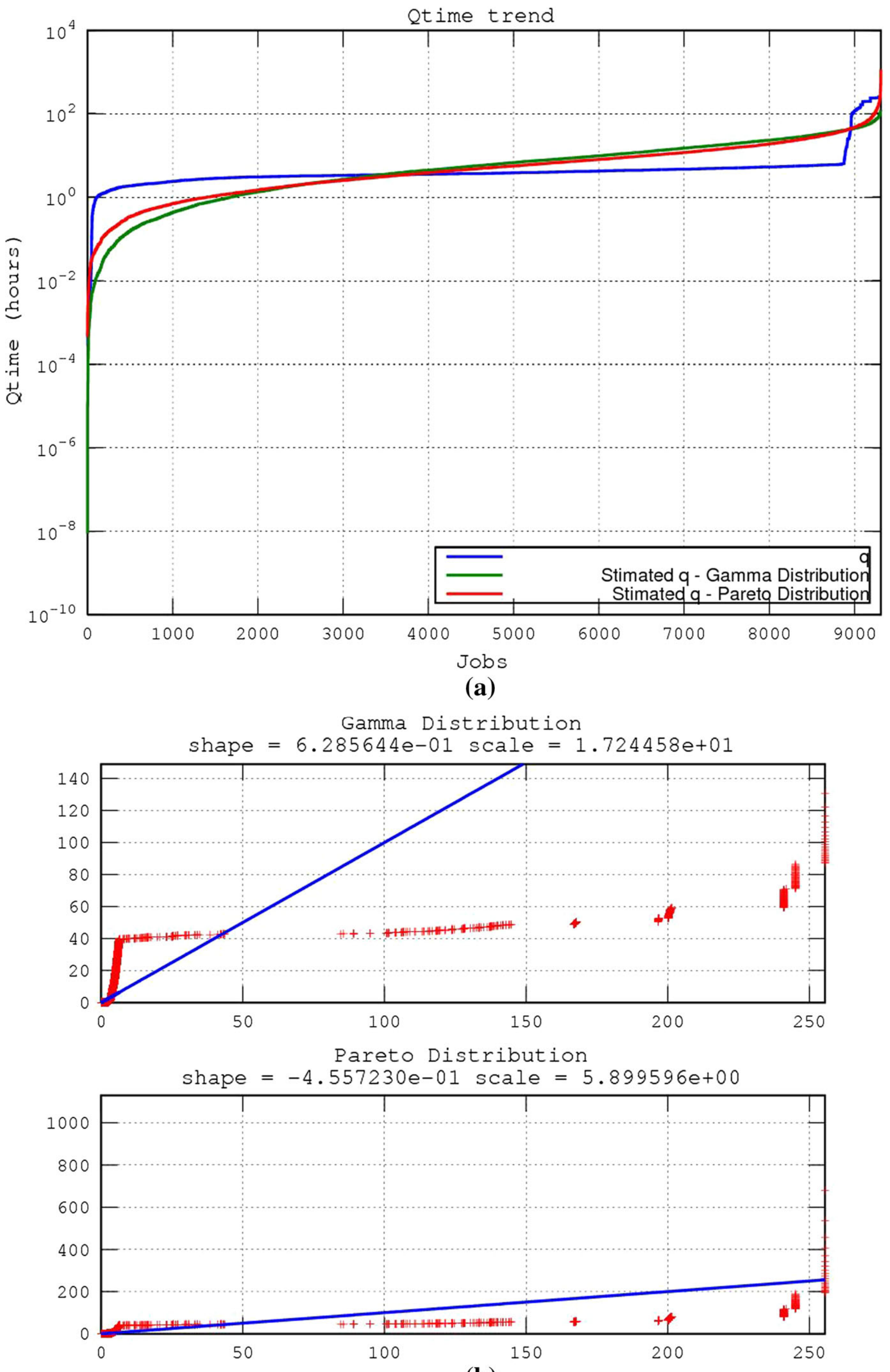


**Fig. 4** Real and estimated queue time trend (**a**) and its probabilistic representation in terms of a Q–Q plot versus both an estimated Pareto distribution and a Gamma distribution (**b**) for medium long sequential jobs during the 20th week of the year 2015

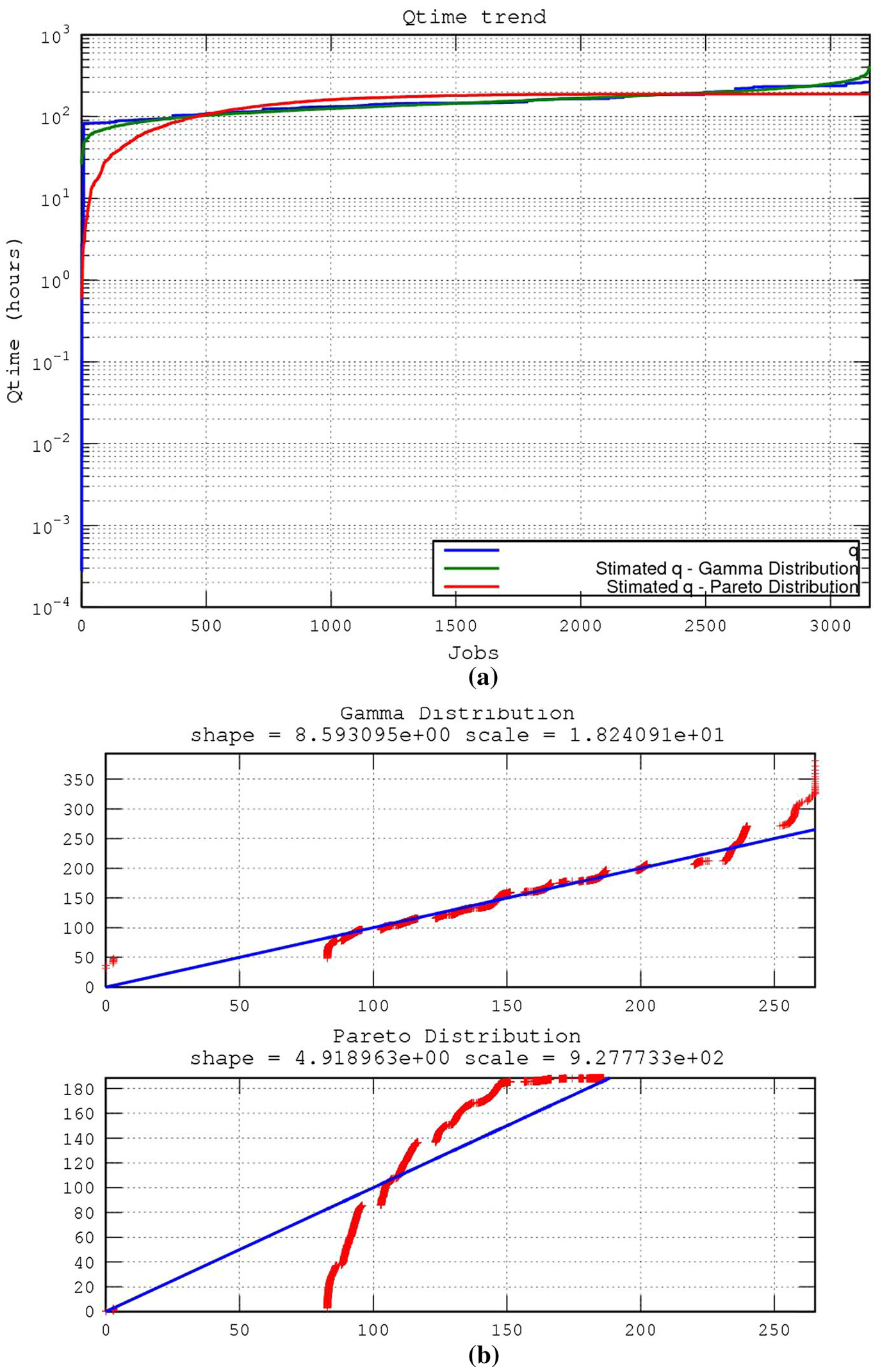
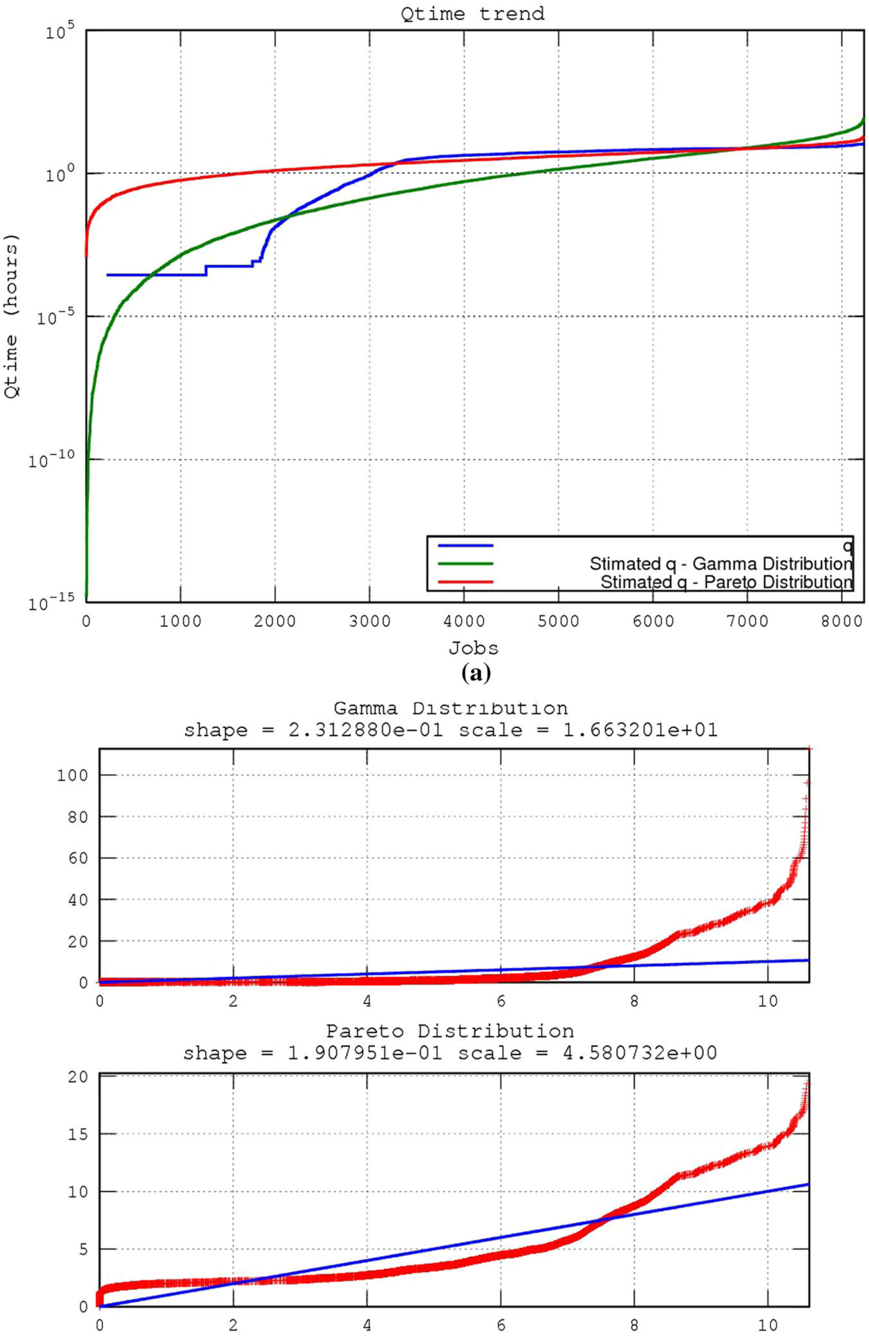


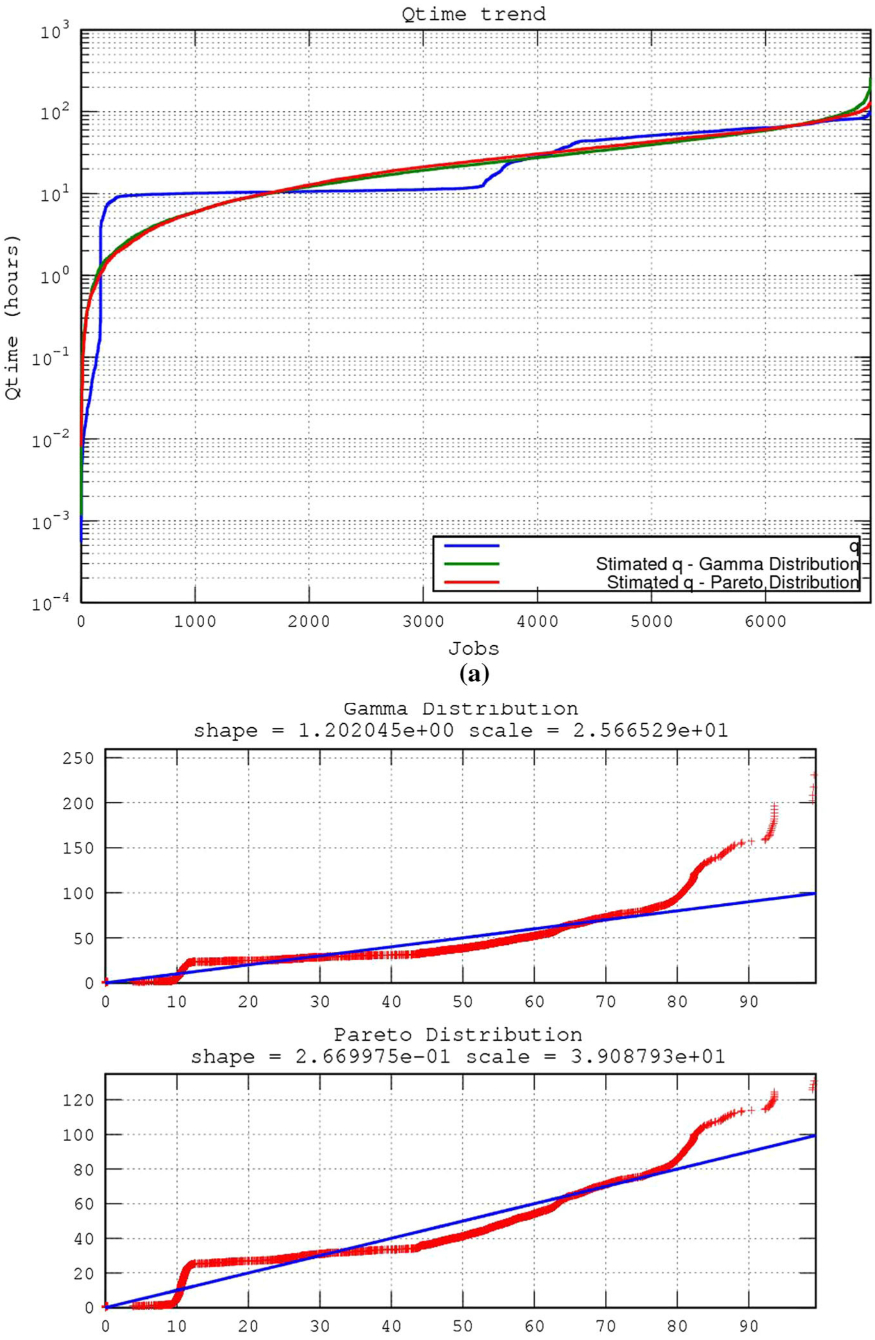
**Fig. 5** Real and estimated queue time trend (**a**) and its probabilistic representation in terms of a Q–Q plot versus both an estimated Pareto distribution and a Gamma distribution (**b**) for medium long sequential jobs during the 26th week of the year 2015



**(a)**

**(b)**

**Fig. 6** Real and estimated queue time trend (**a**) and its probabilistic representation in terms of a Q–Q plot versus both an estimated Pareto distribution and a Gamma distribution (**b**) for medium long sequential jobs during the 28th week of the year 2015



**(a)**

**(b)**

**Fig. 7** Real and estimated queue time trend (**a**) and its probabilistic representation in terms of a Q–Q plot versus both an estimated Pareto distribution and a Gamma distribution (**b**) for medium long sequential jobs during the 40th week of the year 2015

**Table 1** $Q$ values for real and estimated queue time for medium long sequential jobs

| | Week | | | |
|---|---|---|---|---|
| | 20 | 26 | 28 | 40 |
| $q$ | 10.839 | 156.745 | 3.846 | 30.850 |
| $q^{GEst}$ | 10.838 | 156.771 | 3.849 | 30.860 |
| $q^{PEst}$ | 10.832 | 156.715 | 3.845 | 30.847 |

**Table 2** $M_k$ values for real and estimated queue time for medium long sequential jobs

| | Week | | | |
|---|---|---|---|---|
| | 20 | 26 | 28 | 40 |
| $q$ | 281.65 | 269.15 | 54.08 | 123.29 |
| $q^{GEst}$ | 161.49 | 443.76 | 128.10 | 274.13 |
| $q^{PEst}$ | 1370.47 | 235.84 | 56.84 | 153.98 |

```
function PERFORMPARAMETERSFORECAST($tForecasted, nt$)
    for $t \leftarrow tForecasted - nt, tForecasted - 1$ do
                    ▷ estimate Pareto distribution parameters from $q_t$ available data
        PARpar$_t$ ← ESTIMATEPARETO($q_t$)
                    ▷ estimate Gamma distribution parameters from $q_t$ available data
        GAMpar$_t$ ← ESTIMATEGAMMA($q_t$)
    end for
                    ▷ Choose a fitting line for all the PARpar$_t$ values
    $PARp$ ← FITTHEDATA(PARpar)
                    ▷ Choose a fitting line for all the GAMpar$_t$ values
    $GAMp$ ← FITTHEDATA(GAMpar)
                    ▷ Compute the value $PARparForecasted$ at $tForecasted$ of the
                    ▷ line fitting the PARpar$_t$ values
    $PARpar$ ← EVALUATE($PARp, tForecasted$)
                    ▷ Compute the value $GAMparForecasted$ at $tForecasted$
                    ▷ of the line fitting the GAMpar$_t$ values
    $GAMpar$ ← EVALUATE($GAMp, tForecasted$)
    if (decision criterion is satisfied for Pareto parameters) then
        $type$ ← Pareto          ▷ Choose Pareto distribution to characterise
                                  ▷ $q_{tForecasted}$
        $params \leftarrow PARparForecasted$
    else
        $type$ ← Gamma           ▷ Choose Gamma distribution to characterise
                                  ▷ $q_{tForecasted}$
        $params \leftarrow GAMparForecasted$
    end if
    return $type, params$
end function
```

**Fig. 8** Queue time forecast algorithm

Pareto and to a Gamma PDFs defined respectively by the following (5) and (6).

$$f_{q_{notwaiting}}(x) = \frac{ac^a}{(x+c)^{a+1}} \tag{5}$$

**Table 3** $Q$ values for real and forecasted queue time for medium long sequential jobs

| | Week | | | | | | |
|---|---|---|---|---|---|---|---|
| | 20 | 26 | | | 28 | | 40 |
| $q$ | 10.8 | 156.7 | | | 3.85 | | 30.8 |
| | | Rgr. | Spl.,Kts.4 | Spl.,Kts.5 | Rgr. | Rgr.,Wgtd. | |
| $q^{GFor}$ | 6.7 | 9.6 | 61.5 | 100.6 | 182.2 | 44.2 | 10.4 |
| $q^{PFor}$ | 6.9 | 14.8 | 35.7 | 45.1 | 150.7 | 73.1 | 17.1 |

**Table 4** $M_k$ values for real and forecasted queue for medium long sequential jobs

| | Week | | | | | | |
|---|---|---|---|---|---|---|---|
| | 20 | 26 | | | 28 | | 40 |
| $q$ | 281.6 | 269.1 | | | 54.1 | | 123.2 |
| | | Rgr. | Spl.,Kts.4 | Spl.,Kts.5 | Rgr. | Rgr.,Wgtd. | |
| $q^{GFor}$ | 122.7 | 201.6 | 708.1 | 997.4 | 722.8 | 470.4 | 114.4 |
| $q^{PFor}$ | 594.4 | 103.7 | 172.2 | 237.4 | 267.0 | 502.6 | 120.6 |

$$f_{q_{waiting}}(y) = \frac{y^{\alpha-1}\exp(-y/\lambda)}{\lambda^{\alpha}\Gamma(\alpha)} \tag{6}$$

Parameters $a$ and $c$ in (5) are named *shape* and *scale* of the Pareto distribution respectively. Parameters $\alpha$ and $\lambda$ in (6) are named *shape* and *scale* of the Gamma distribution respectively.

Assuming that the work flow (or its subset) acting on the system is characterised by a huge set of jobs with a close to zero waiting time, then we can assume that the random variable $q_{notwaiting}$ dominates $q_{waiting}$ in the sense described in [10]. Vice versa, if all the jobs (or most of them) have a not null waiting time, we can assume that $q_{waiting}$ dominates $q_{notwaiting}$.

All above said, depending on how the work flow can be characterised (primarily as consisting of a very large number of jobs which not have to wait—respectively, have to wait—in the queue), we assume that the variable $q$ can be better characterised as a random variable with distribution of type Pareto (respectively, Gamma).

To experimentally validate our assumptions, we performed some tests using one of the jobs classes represented in Fig. 1. In particular, we focus our attention on the jobs belonging to the *T1P0* class constituted by sequential jobs with a medium execution time (it ranges from 2 h to 2 days). We choose these jobs since they are a substantial part of the entire work flow and, thus, they represent a statistically significant sample.

The validation process is performed by the following steps:

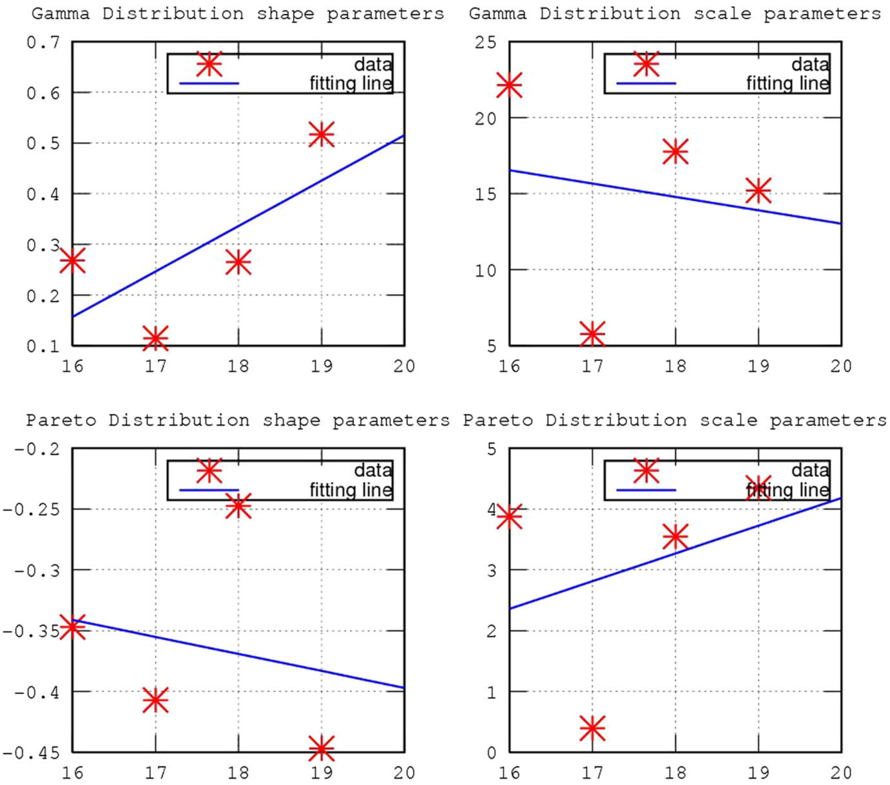


**Fig. 9** Trend and forecast (by using a linear regression approach) of distribution parameters for medium long sequential jobs during the 20th week of the year 2015

1. Fixed the reference time period of 1 week, we collected the values of queue time $q$ for all the jobs in *T1P0* class terminated during the reference time period;
2. We estimated the values of $\alpha$ and $\lambda$ parameters starting from the values of $q$ assuming that they are distributed as a Gamma distribution;
3. We estimated the values of $a$ and $c$ parameters starting from the values of $q$ assuming that they are distributed as a Pareto distribution;
4. We compared, using a quantile–quantile (Q–Q) plot, the trends of the quantile of the *"estimated"* distribution identified at the above point 2 and the trends of the quantile of $q$;
5. We compared, using a quantile–quantile (Q–Q) plot, the trends of the quantile of the *"estimated"* distribution identified at the above point 3 and the trends of the quantile of $q$.

Parameters estimation can be performed by classical statistical methods such as the *"method of moments"* and the *"maximum likehood estimation"* [20]. A quantile–quantile (Q–Q) plot is often used to see if a given set of data belongs to a specified distribution. It should be approximately linear if such distribution is the correct model

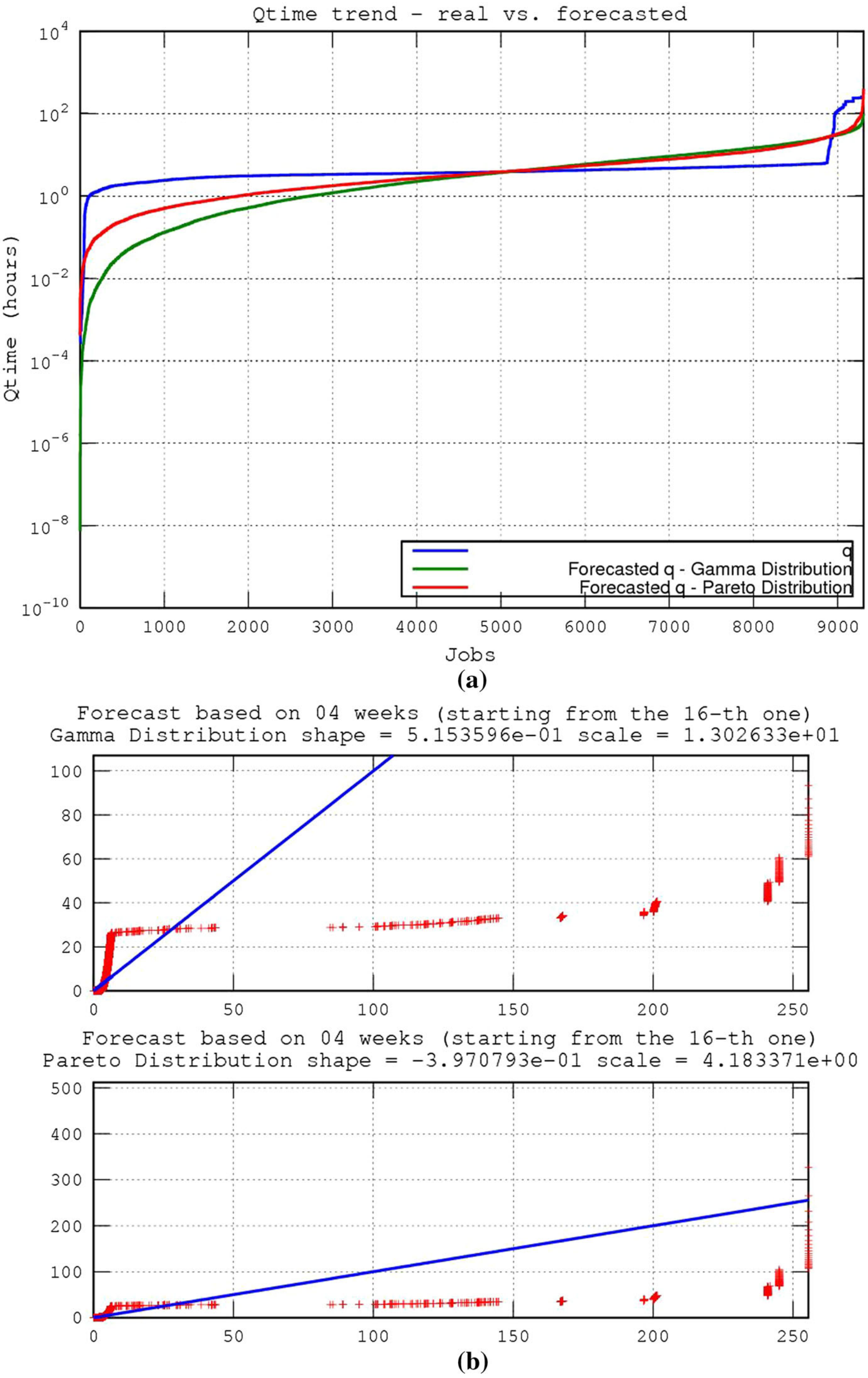
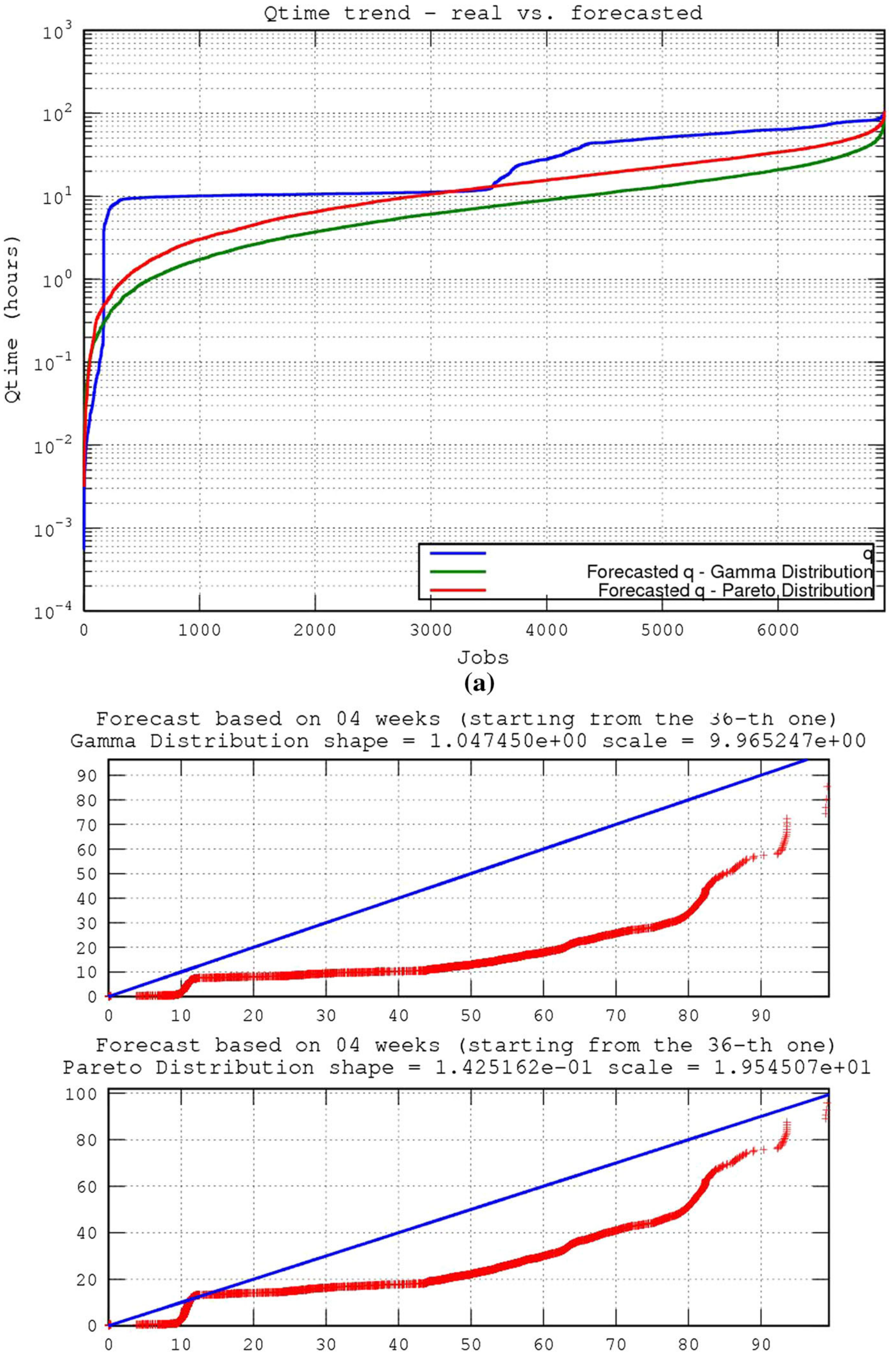


**Fig. 10** Real and forecasted queue time trend (**a**) and Q–Q plot of real versus forecasted (by using a linear regression approach) queue time (**b**) for medium long sequential jobs during the 20th week of the year 2015



**(a)**

**(b)**

**Fig. 11** Real and forecasted queue time trend (**a**) and Q–Q plot of real versus forecasted (by using a linear regression approach) queue time (**b**) for medium long sequential jobs during the 40th week of the year 2015

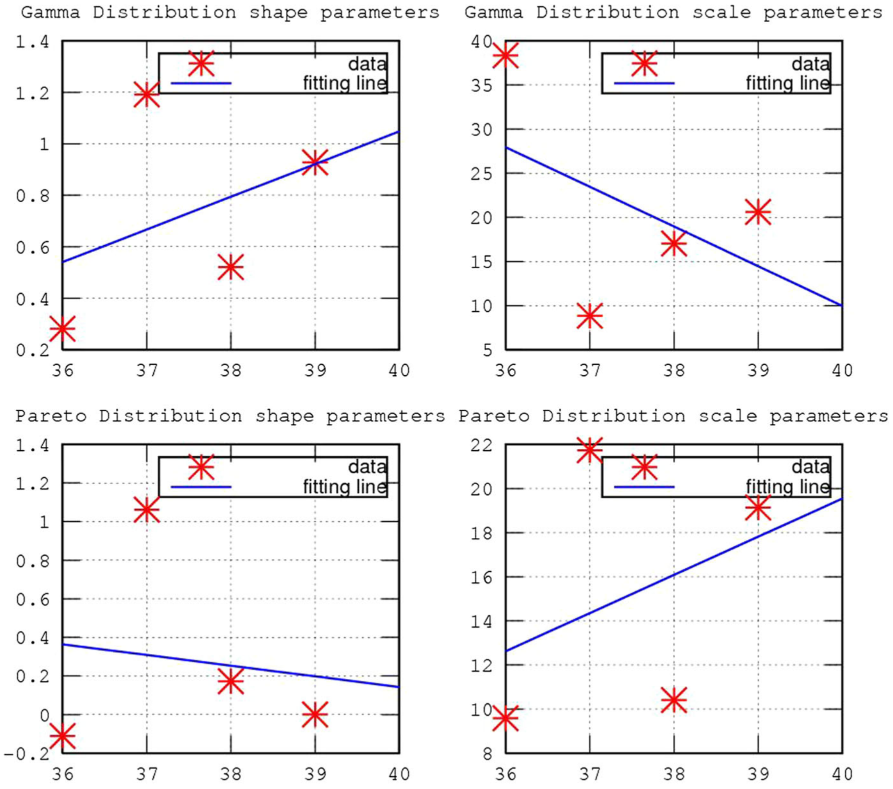


**Fig. 12** Trend and forecast (by using a linear regression approach) of distribution parameters for medium long sequential jobs during the 40th week of the year 2015

(e.g., see [23]). In Fig. 3 we plot trends of the estimated values for $\alpha$ and $\lambda$ and for $a$ and $c$ related to $q$ in all the weeks of the 2015 year. Note the strong variation of these parameters as a symptom of the strong variability of $q$ during the past year.
Here and in the next subsection we will focus on:

- Weeks preceded by regular behaviour for the parameters trend (e.g., the 20th and 40th weeks),
- Weeks corresponding to a significant peak (e.g., the 26th week):
- Weeks preceded by at least a week with a significant peak (e.g., the 28th week).

In Figs. 4a, 5a, 6a and 7a we show the values of $q$, $q^{PEst}$ and $q^{GEst}$ for all the jobs terminated during the 20th, the 26th, the 28th and the 40th weeks of the year 2015. With the symbols $q^{PEst}$ and $q^{GEst}$ we represent values generated from the distributions identified at the above points 3 and 2 respectively. In Figs. 4b, 5b, 6b and 7b we show the trends of the Q–Q plots obtained at the points 4 and 5 above for all the jobs terminated respectively during the same weeks.

From Fig. 4, we can observe that when the number of jobs with null waiting time is very high, $q$ is well represented by a Pareto distribution. Vice versa when all jobs wait

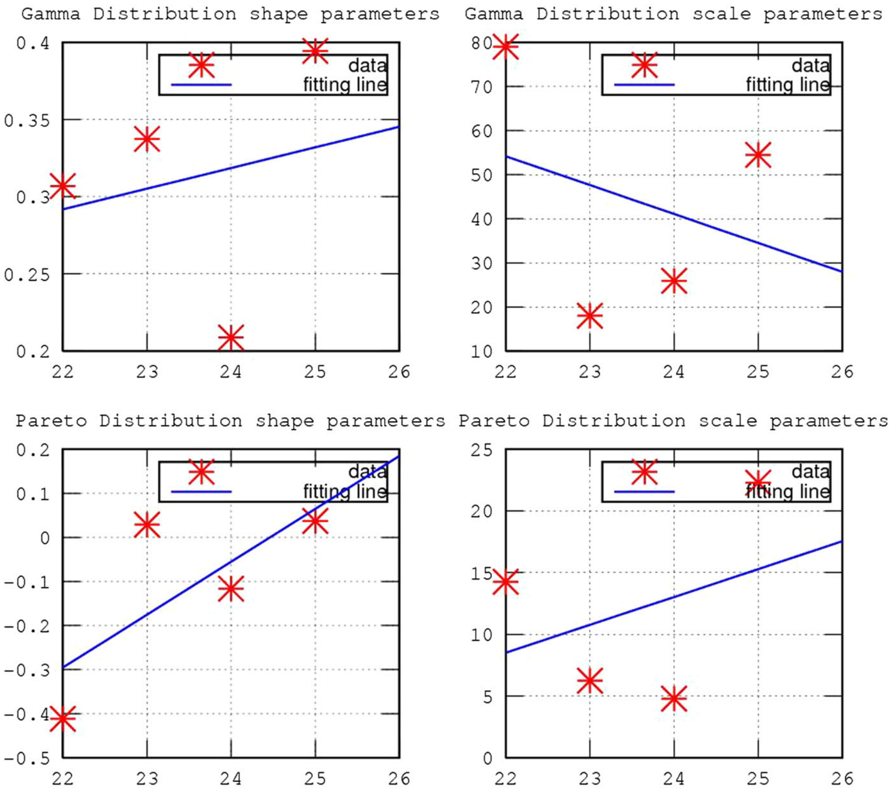


**Fig. 13** Trend and forecast (by using a linear regression approach) of distribution parameters for medium long sequential jobs during the 26th week of the year 2015

some time (see Fig. 5), $q$ is very well represented by a Gamma distribution. Both the representations are affected by issues related with the maximum values for $q^{PEst}$ and $q^{GEst}$. From Fig. 6 and 7 we can argue that, in *hybrid* scenarios, $q$ can be represented well enough by both the distributions. From Tables 1 and 2 we can also observe that the *"estimated"* distributions generate numbers $q^{PEst}$ and $q^{GEst}$ useful to compute values of $Q$ and $M_k$ very close to those ones computed from the real values of $q$ (in some cases values of $M_k$ are very overestimated because of the maximum values of $q^{PEst}$ and $q^{GEst}$).

### 5.2 Queue Time Forecast by Means of Probabilistic Characterisation and Data Fitting

The probabilistic characterisation of the queue time $q$ can be useful as a tool that, combined with other techniques, provides a short term forecast of queue time itself starting from the queue time estimated in some $nt$ time references in the past

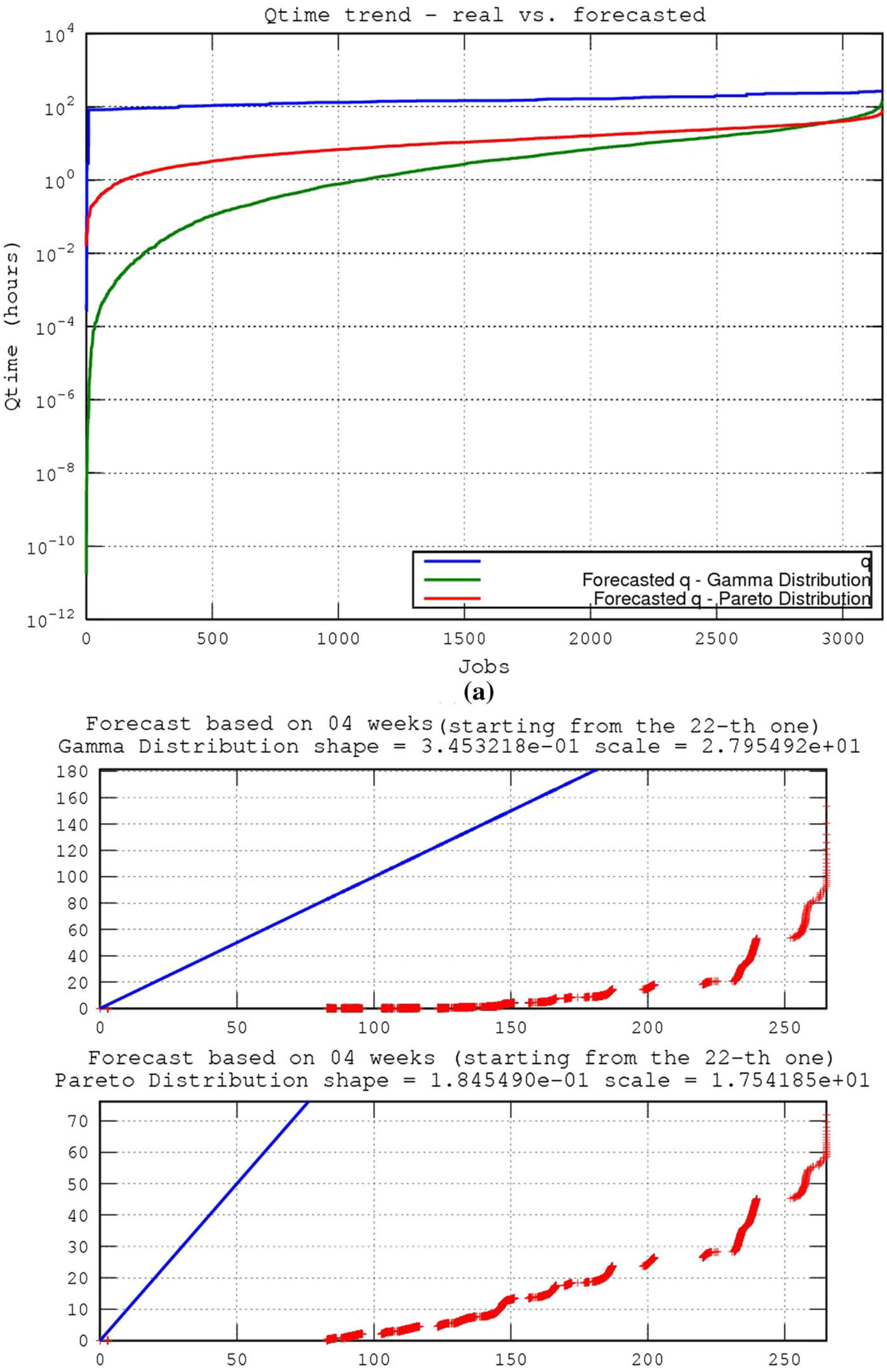
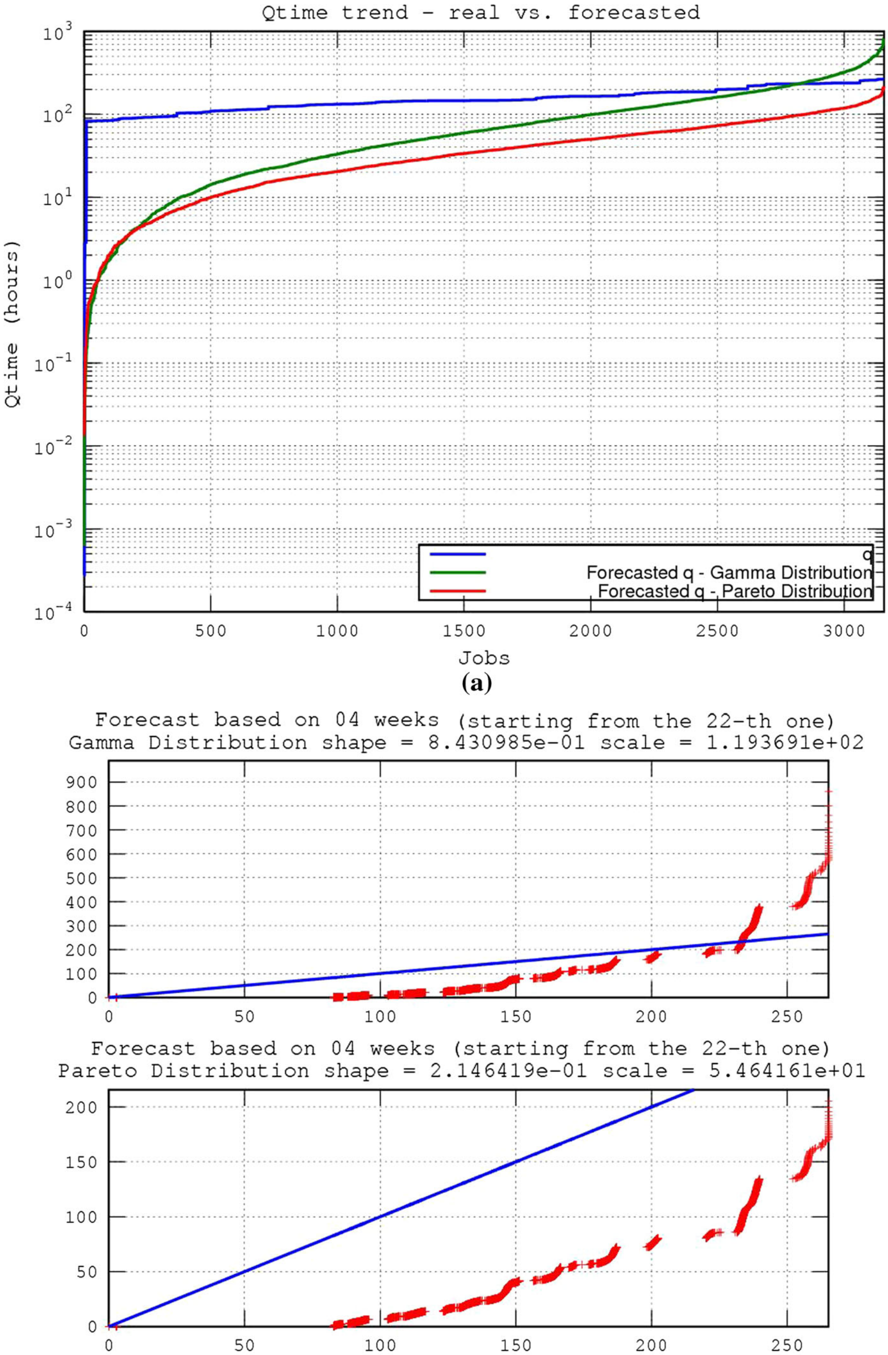


**Fig. 14** Real and forecasted queue time trend (**a**) and Q–Q plot of real versus forecasted (by using a linear regression approach) queue time (**b**) for medium long sequential jobs during the 26th week of the year 2015



**(a)**

**(b)**

**Fig. 15** Real and forecasted queue time trend (**a**) and Q–Q plot of real versus forecasted (by using a spline based approach—$knots = 5$, $order = 1$) queue time (**b**) for medium long sequential jobs during the 26th week of the year 2015

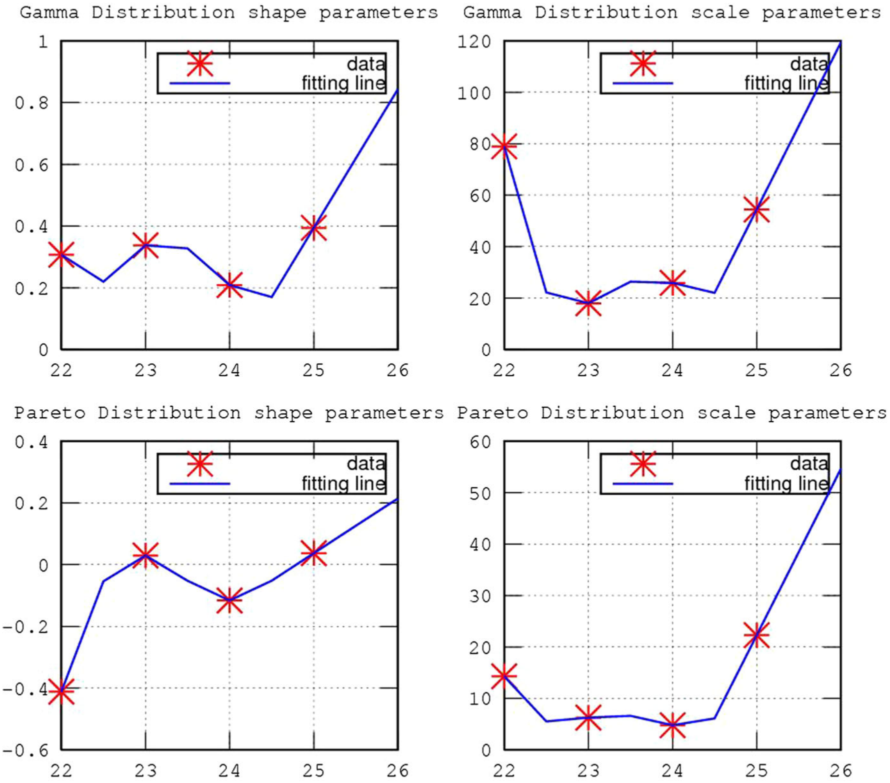


**Fig. 16** Trend and forecast (by using a spline based approach—$knots = 5$, $order = 1$) of distribution parameters for medium long sequential jobs during the 26th week of the year 2015

$$\{q\,(t)\}_{t=tForecasted-nt,\ldots,tForecasted-1}\,.$$

We use the algorithm in Fig. 8 to estimate $q\,(tForecasted)$ (the queue time to be forecasted). The algorithm 8 combines numerical analysis techniques, as the fitting of data, with statistical tools as the parameters estimation.

We validated the algorithm through some tests aimed, essentially, to verify the quality of the parameters $PARpar$ and $GAMpar$ (see lines 14 and 17 of the algorithm in Fig. 8) *"forecasted"* (Figs. 9, 12, 13, 16, 17, 20, 21) for both the considered distributions by means of:

- The trends comparison for the values of $q^{GEst}$ and $q^{PEst}$ generated by the distributions *"estimated"* from the data $q$ and the values of $q^{GFor}$ and $q^{PFor}$ whose parameters have been *"forecasted"* by the algorithm in Fig. 8 [see Figs. (4, 5, 6, 7)a vs. (10, 11, 14, 15, 18, 19, 22)a];
- The trends comparison for the Q–Q plot related to $q^{GEst}$ and $q^{PEst}$ with $q^{GFor}$ and $q^{PFor}$ [see Figs. (4, 5, 6, 7)b vs. (10, 11, 14, 15, 18, 19, 22)b];

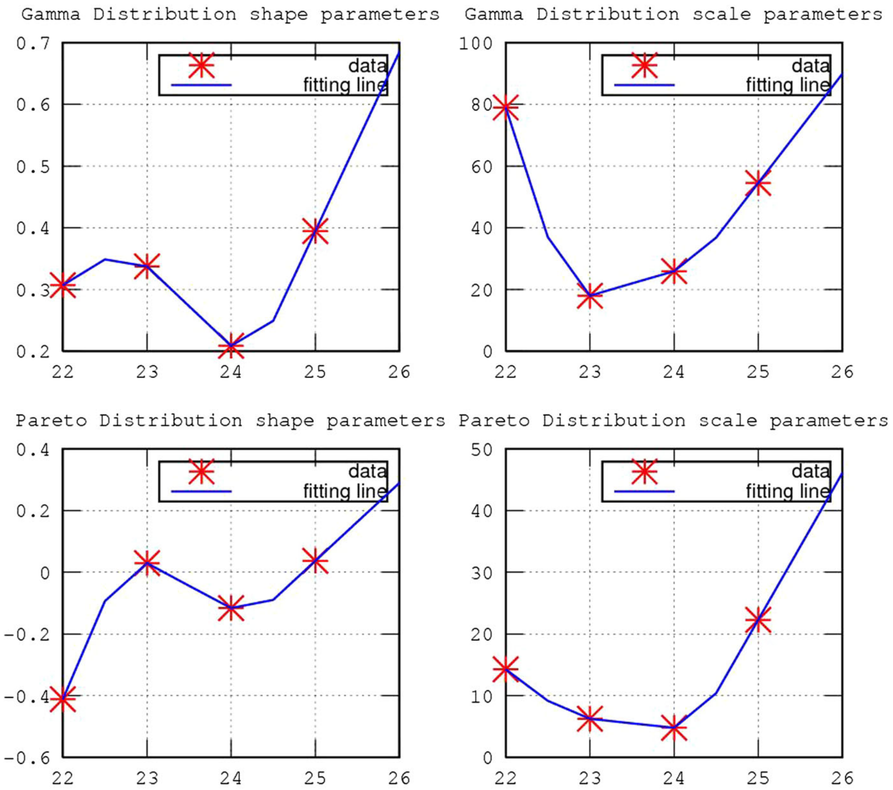


**Fig. 17** Trend and forecast (by using a spline based approach—$knots = 4$, $order = 1$) of distribution parameters for medium long sequential jobs during the 26th week of the year 2015

- The comparison of the values for the key-statistics $Q$ and $M_k$ computed from $q^{GEst}$ and $q^{PEst}$ with $q^{GFor}$ and $q^{PFor}$ (see Tables 1, 2 vs. 3, 4).

The reference time period constists in 1 week. The considered weeks are the same used for the tests described in previous Sect. 5.1 where the time references in the past is $nt = 4$. These weeks are representative of the following four possible real scenarios:

*Weeks nr. 20 and 40* The trend of the parameter values is quite regular (see Fig. 3) in the time interval

$$[tForecasted - nt, \ldots, tForecasted].$$

In a such scenario a fitting technique, based on the linear regression (see Figs. 9, 12), appears to be working well enough (see Q–Q plots in Figs. 10b and 4b or 11b and 7b). Moreover, the fitting technique used seems to have *regularising properties* in parameters values calculation since distributions identified by the latter seem to best describe $q$ than those ones identified by estimating the same parameters from $q$ do

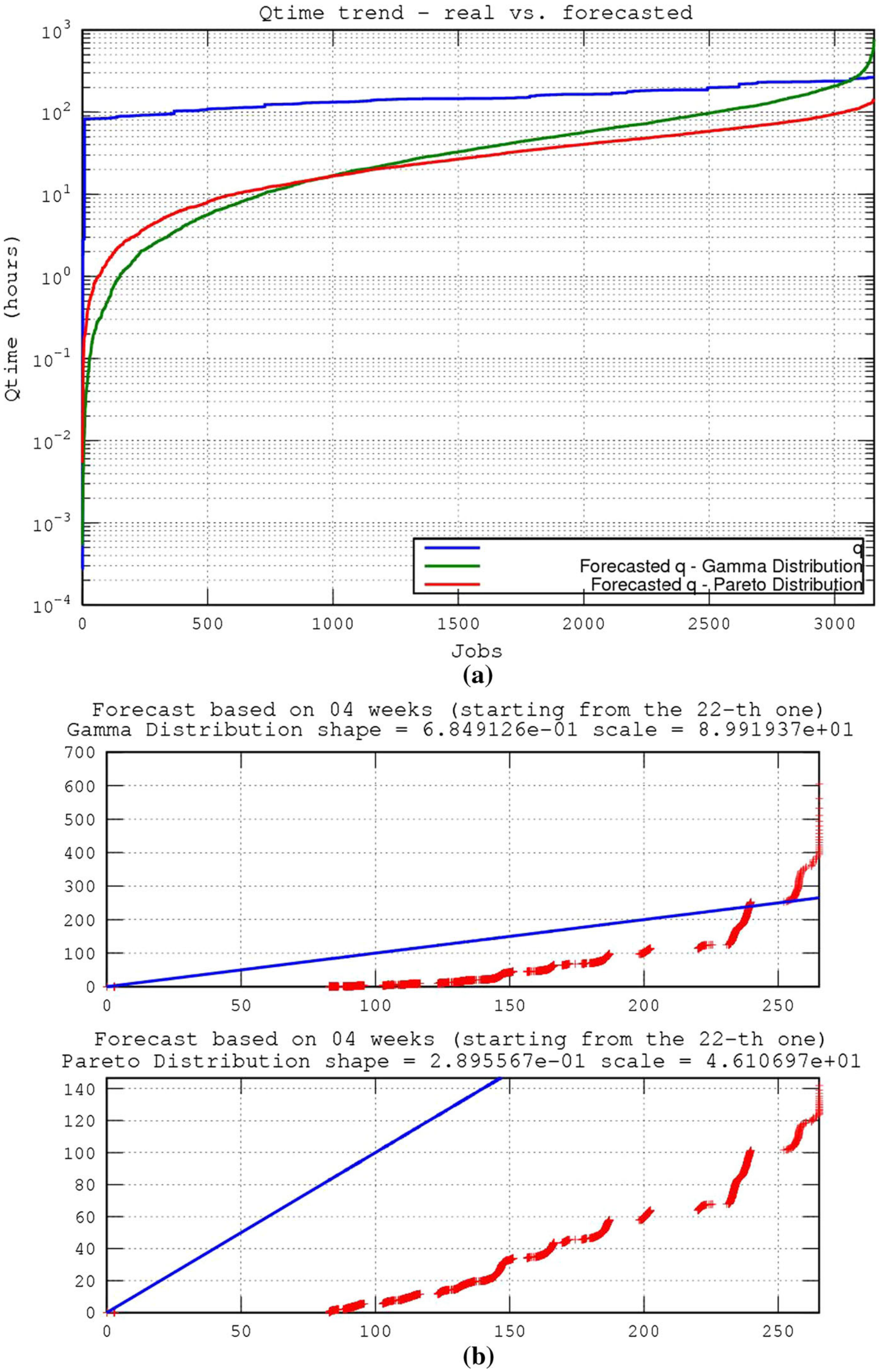


**Fig. 18** Real and forecasted queue time trend (**a**) and Q–Q plot of real versus forecasted (by using a spline based approach—$knots = 4$, $order = 1$) queue time (**b**) for medium long sequential jobs during the 26th week of the year 2015

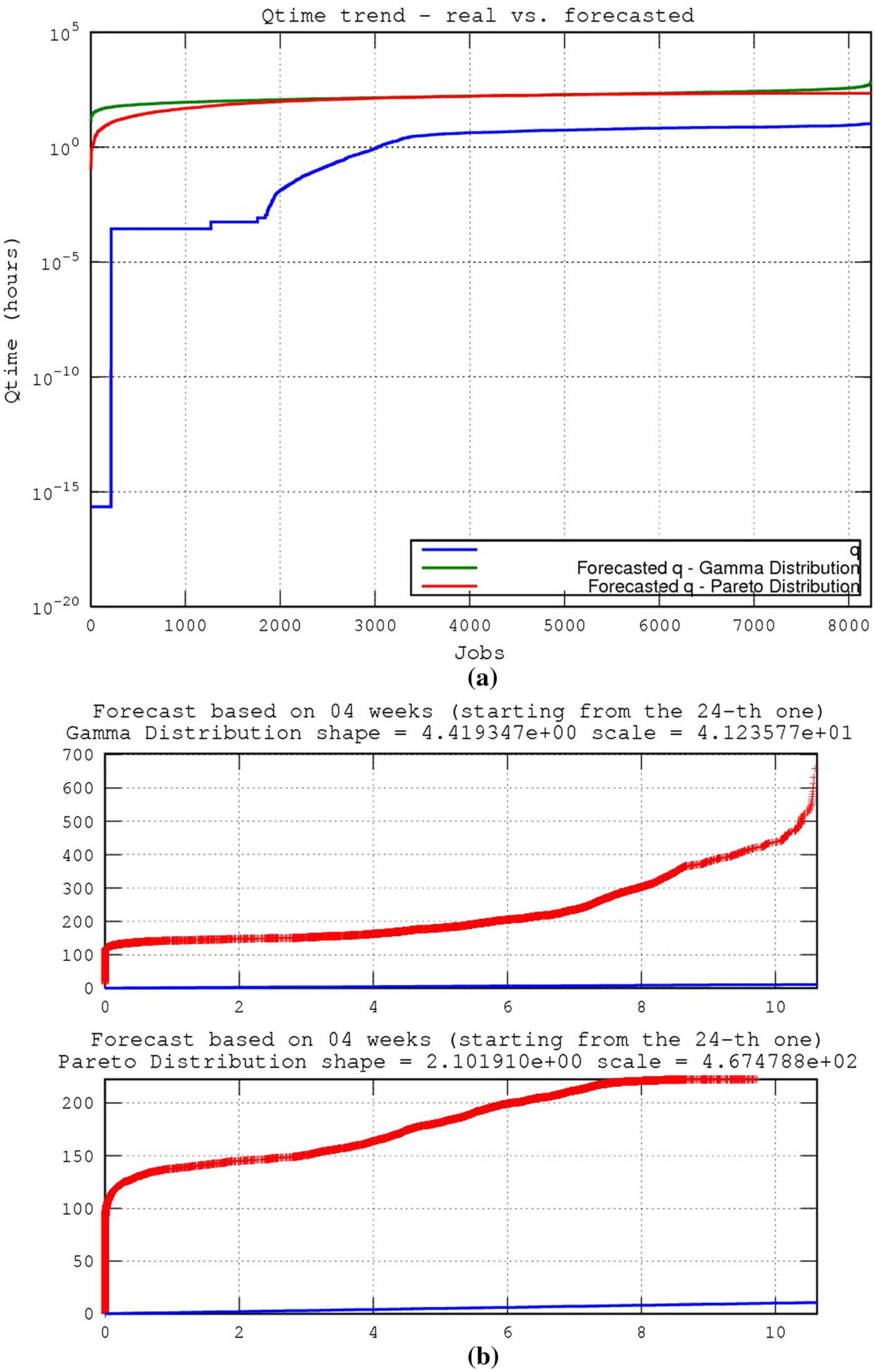


**Fig. 19** Real and forecasted queue time trend (**a**) and Q–Q plot of real versus forecasted (by using a linear regression approach) queue time (**b**) for medium long sequential jobs during the 28th week of the year 2015

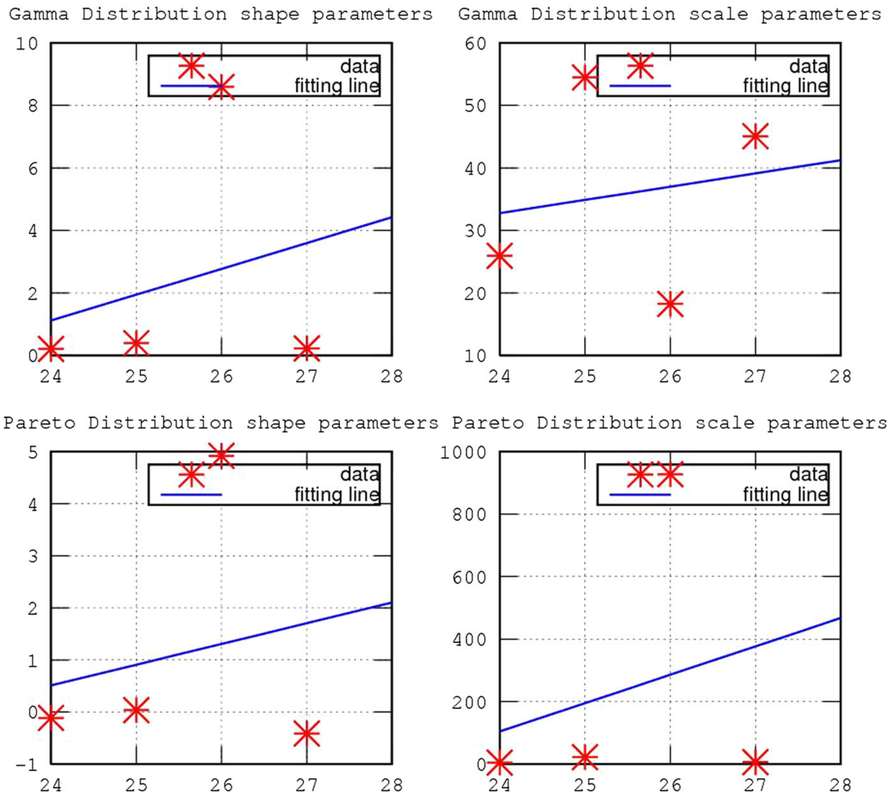


**Fig. 20** Trend and forecast (by using a linear regression approach) of distribution parameters for medium long sequential jobs during the 28th week of the year 2015

(see $q$ trends in Figs. 10a and 4a or 11a and 7a). The *regularising properties* seem to have beneficial effects also for a more correct estimate of $M_k$ values (see Tables 2, 4) *"smoothing"* the contribution from the maximum values of $q^{GFor}$ and $q^{GEst}$.

*Week nr. 26* The trend of the parameter values is quite regular in the time interval

$$[tForecasted - nt, \ldots, tForecasted - 1],$$

but it presents a significant peak for $t = tForecasted$ (see Fig. 3). It represents the worst possible scenario: here, only non-linear fitting techniques (for example those based on spline, see Figs. 13, 17, 16) are able to predict accurate enough values for the parameters (see comparison of Q–Q plots in Figs. 14b with 5b, 15b with 5b and 18b with 5b).

*Week nr. 28* The trend of the parameter values is not regular in the time interval

$$[tForecasted - nt, \ldots, tForecasted]$$

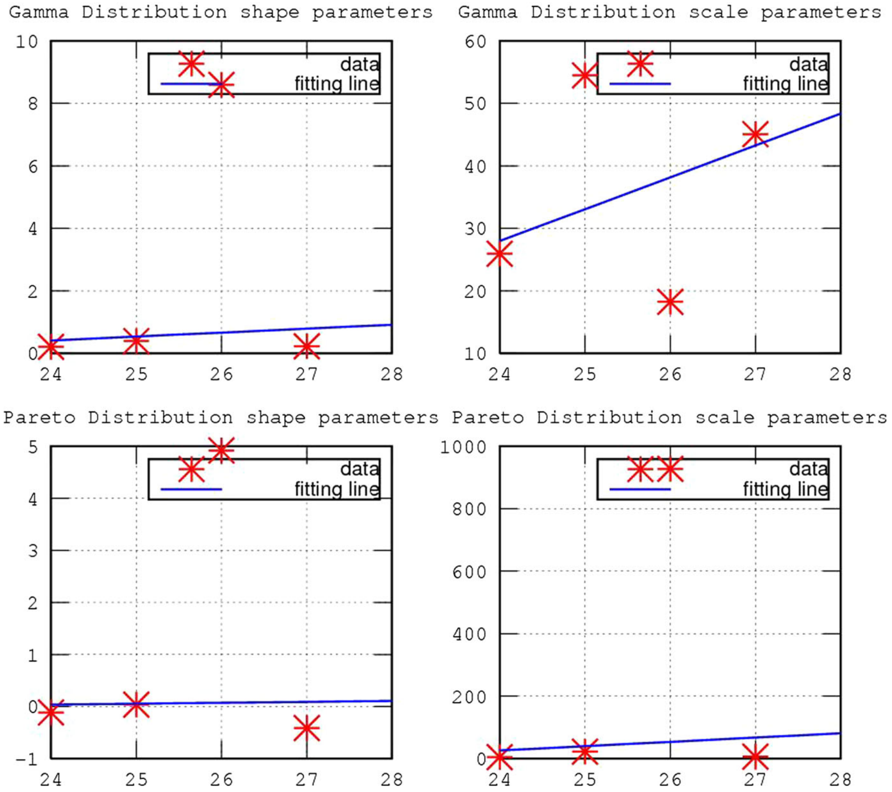


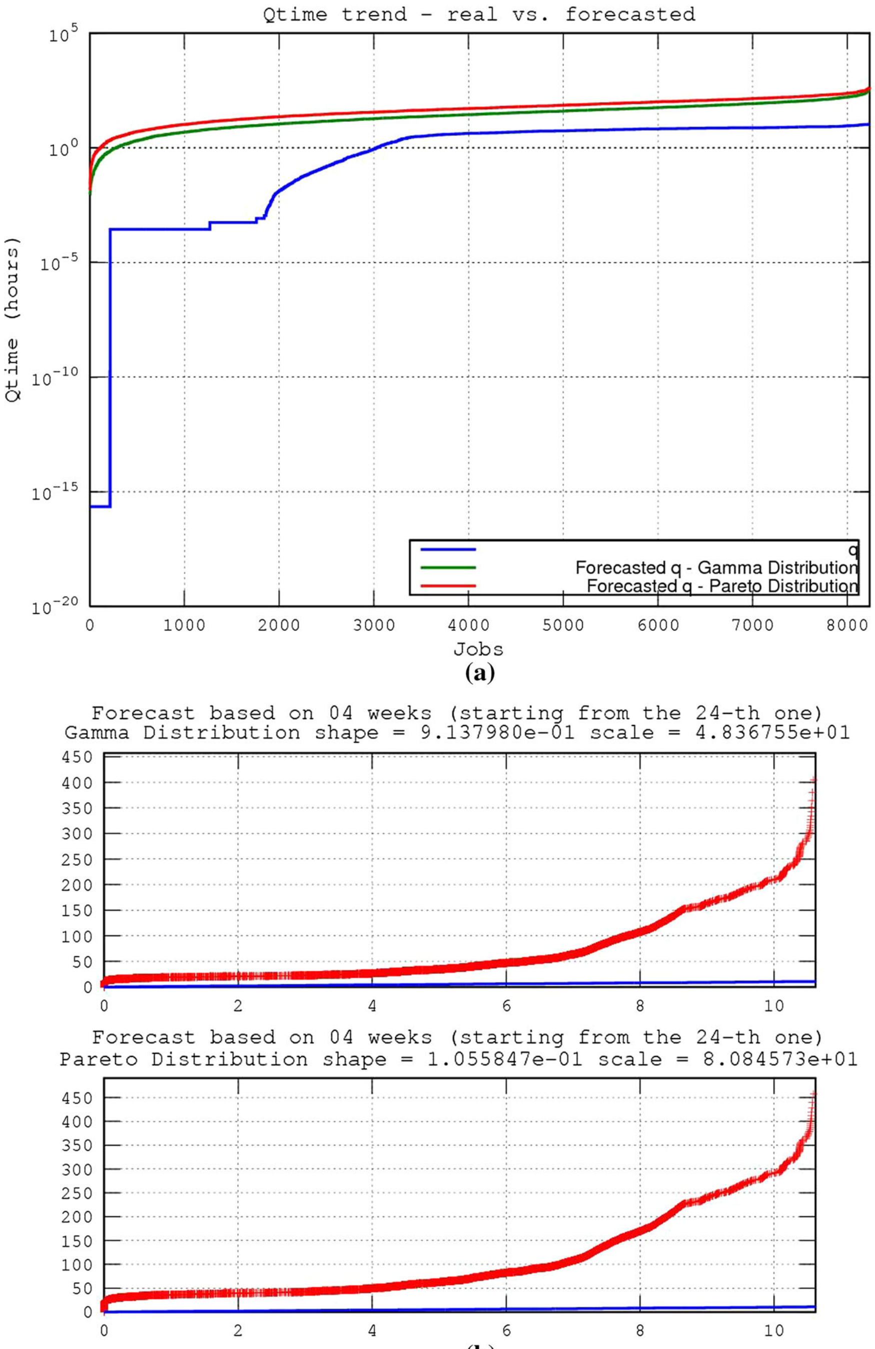
**Fig. 21** Trend and forecast (by using a weighted linear regression approach) of distribution parameters for medium long sequential jobs during the 28th week of the year 2015

because of the presence of an internal significant peak (see Fig. 3). In this scenario, a fitting technique based on weighted (see Fig. 21) linear regression (see Q–Q plots in Figs. 6b, 22b) partially mitigates the problems related to the use of the classic linear regression (see Fig. 20) (see Q–Q plots in Figs. 6b, 19b). Unfortunately, the estimated values of $Q$ and $M_k$ are quite unsatisfactory.

## 6 Conclusion and Future Work

In this document, we described the progresses made to devise ASC, which aims to gain a balanced, efficient and effective use of computing resources by heterogeneous users communities. Here, we gave details about an approach to find the most suitable PDF able to characterise and to forecast queue time. The final aim is the chance to have a computable estimation of the considered *key-statistics*.

Some preliminary results are presented, validated on the SCoPE use case. We implemented an embryonal algorithm that computes both queue time estimation and forecast. We still have to work on refining the proposed approach at the aim to:



**(a)**

**(b)**

**Fig. 22** Real and forecasted queue time trend (**a**) and Q–Q plot of real versus forecasted (by using a weighted linear regression approach) queue time (**b**) for medium long sequential jobs during the 28th week of the year 2015

- Build a *decision criterion* to choose *the best* distribution describing the data (see line 18 of the algorithm in Fig. 8);
- Build a *decision criterion* to choose *the best* fitting method of the historical parameters (see lines 9 and 11 of the algorithm in Fig. 8).

We also are working to improve the overall quality of $Q$ and $M_k$ values computed from the forecasted queue time.

**Acknowledgements** This work is part of the activities of a multidisciplinary group (GTT), responsible for the SCoPE infrastructure management. It has been realised thanks to the use of the SCoPE computing infrastructure at the University of Naples, also in the framework of PON *”Rete di Calcolo per SuperB e le altre applicazioni”* (ReCaS) project.

## References

1. Aguilar, J., Gelenbe, E.: Task assignment and transaction clustering heuristics for distributed systems. Inf. Sci. **97**, 1–2 (1997)
2. Barone, G., Boccia, V., Bottalico, D., Campagna, R., Carracciuolo, L., Laccetti, G.: An approach to model resources rationalisation in hybrid clouds through users activity characterisation. In: Proceedings of Future Computing 2015—The Seventh International Conference on Future Computational Technologies and Applications, Proceedings of FUTURE COMPUTING, International Conference on Future Computational Technologies and Applications, pp. 48–53 (2015). http://www.thinkmind.org/download.php?articleid=future_computing_2015_3_10_30022
3. Barone, G., Boccia, V., Bottalico, D., Carracciuolo, L., Doria, A., Laccetti, G.: Modelling the behaviour of an adaptive scheduling controller. In: 2012 Sixth International Conference on Complex, Intelligent and Software Intensive Systems (CISIS), pp. 438–442 (2012). doi:10.1109/CISIS.2012.26
4. Caruso, P., Laccetti, G., Lapegna, M.: A performance contract system in a grid enabling, component based programming environment. In: Advances in Grid Computing—EGC 2005, European Grid Conference, Amsterdam, The Netherlands, February 14–16, 2005, Revised Selected Papers, pp. 982–992 (2005). doi:10.1007/11508380_100
5. Casavant, T., Kuhl, J.: A taxonomy of scheduling in general-purpose distributed computing systems. IEEE Trans. Softw. Eng. **14**(2), 141–154 (1988). doi:10.1109/32.4634
6. Federated cloud: Building a grid of clouds. https://www.egi.eu/. Accessed 31 July 2016
7. Foster, I.: What’s faster a supercomputer or EC2? (2009). http://ianfoster.typepad.com/blog/2009/08/whats-fastera-supercomputer-or-ec2.html. Accessed 31 July 2016
8. Gkoutioudi, K., Karatza, H.D.: Task cluster scheduling in a grid system. Simul. Model. Pract. Theory **18**(9), 1242–1252 (2010)
9. Italian Grid Infrastructure: http://www.italiangrid.it/. Accessed 31 July 2016
10. James Allen Fill, M.M.: Stochastic monotonicity and realizable monotonicity. Ann. Probab. **29**(2), 938–978 (2001)
11. Laccetti, G., Lapegna, M., Mele, V., Romano, D., Murli, A.: A double adaptive algorithm for multidimensional integration on multicore based HPC systems. Int. J. Parallel Program. **40**(4), 397–409 (2012). doi:10.1007/s10766-011-0191-4
12. Merola, L.: on behalf of S.Co.P.E. Project: The S.Co.P.E. Project. In: Proceedings of the Final Workshop of the Grid Projects of the Italian National Operational Programme 2000–2006 Call 1575, Edited by Consorzio COMETA, pp. 18–35 (2008)
13. Murli, A., Boccia, V., Carracciuolo, L., D’Amore, L., Laccetti, G., Lapegna, M.: Monitoring and migration of a PETSc-based parallel application for medical imaging in a grid computing PSE. IFIP Int. Fed. Inf. Process. **239**, 421–432 (2007). doi:10.1007/978-0-387-73659-4_25
14. Nadarajah, S.: The waiting time distribution. Comput. Ind. Eng. **53**(4), 693–699 (2007). doi:10.1016/j.cie.2007.06.004
15. Nazir, A., Srensen, S.A.: Cost-benefit analysis of high performance computing infrastructures. In: SOCA, pp. 1–8. IEEE (2010)
16. Papazachos, Z.C., Karatza, H.D.: The impact of task service time variability on gang scheduling performance in a two-cluster system. Simul. Model. Pract. Theory **17**, 1276–1289 (2009)

17. Potts, C.N., Strusevich, V.A.: Fifty years of scheduling: a survey of milestones. JORS (2009). doi:10.1057/jors.2009.2
18. Powers, S.: A study of the impact of scheduling parameters in heterogeneous computing environments. In: Proceedings of the 2014 Winter Simulation Conference. WSC ’14, pp. 933–942. IEEE Press, Piscataway (2014)
19. Serazzi, G., Calzarossa, M.: Adaptive optimization of a system’s load. IEEE Trans. Softw. Eng. **10**(6), 837–845 (1984)
20. STAT 415 Intro Mathematical Statistics online course. https://onlinecourses.science.psu.edu/stat414/node/3. Accessed 31 July 2016
21. Sun, H., Cao, Y., Hsu, W.J.: Fair and efficient online adaptive scheduling for multiple sets of parallel applications. In: ICPADS, pp. 64–71. IEEE (2011)
22. Sztrik, J.: Queueing theory and its applications: a personal view. In: Proceedings of the Third Symposium on Information and Communication Technology, SoICT ’12, pp. 1–1. ACM, New York (2012). doi:10.1145/2350716.2350717
23. Wilk, M.B., Gnanadesikan, R.: Probability plotting methods for the analysis of data. Biometrika **55**(1), 1–17 (1968). doi:10.1093/biomet/55.1.1