Efficient resubmission strategies to design robust grid production environments

Diane Lingrand and Johan Montagnat
University of Nice - Sophia Antipolis / CNRS
Sophia Antipolis, FRANCE
Email: {lingrand,johan}@i3s.unice.fr

Abstract—Production grids exhibit high failure rates hampering the development of many large scale scientific applications. End users require robust experiment production environments ensuring efficient resubmission of failed tasks. Proper parameterization of resubmission strategies is a complex problem that depends on the non-stationary workload conditions experienced by the infrastructure. In order to determine optimal resubmission parameters, probabilistic models of the overhead experienced by grid jobs are defined, taking into account the distribution of faults as measured on the infrastructure. Two strategies that can be implemented on the client side are proposed. Their models are evaluated under variable workload conditions to assess their validity along time. Their results are compared and a trade-off between usability and model accuracy is discussed.

Keywords-Grid computing; Fault tolerance; Probabilistic modeling.

I. INTRODUCTION

Although increasingly adopted by diverse scientific communities, production grid infrastructures are still considered complex by end-users. The counterpart of the scalability and collaborative capabilities of grids is the significant investment needed to build reliable distributed applications. Despite best efforts in middleware development, one of the major difficulties encountered by end-users is the high failure rate observed on grids. It is inherent to any large scale distributed system, due to the interdependencies between many software components, the unprecedented amount of computing resources involved, the heterogeneity of middleware stacks, the concurrent exploitation by many users, and the criticality of communications between distributed components. Robust experiment production environments, centered on the client side to be resilient against connectivity problems, are therefore mandatory for most usages.

The most basic functionality needed by all end-users is the capability to resubmit failed tasks to ensure experiments completion. Failure solving and resubmission should not be let under the responsibility of the user. It is an extremely tedious and error-prone process when dealing with large scale experiments. Furthermore, end-users are lacking expertise and information to make proper decisions. Indeed, the behavior of production grids is difficult to comprehend, considering their non stationary workload and the interconnection of many distributed entities. Since it is hardly feasible to obtain a fine-grained model of a running production grid

infrastructure, global probabilistic models of grid workloads are increasingly used [1], [2], [3] to tackle the complexity of grid systems. Large collections of real production grid usage traces are collected for further analysis nowadays [4], [5], [6]. These traces exhibit large number of *failed jobs* (jobs that cannot complete execution due to a specific problem reported by the middleware) and *outliers* (jobs for which no trace information is reported by the system after some error happened). For instance, analyzing more than 33 millions of job traces collected on the EGEE European production grid infrastructure[1] over 22 months (September 2005 to June 2007) shows as much as 19% of failed jobs and 16% of outliers. With about 35% of jobs not completing normal execution. Furthermore, even successful jobs experience variable *latency* (time between job submission and jobs starting execution), typically characterized by heavy-tailed probability laws [1], [3]. As a consequence, a non-negligible number of jobs face highly penalizing overheads in any large scale experiment. Resubmission of delayed jobs [8] or more aggressive multi-submission strategies [7], [8] can significantly improve grid usage experience.

This work aims at developing models of job resubmission strategies that can be implemented on the client side in production grid systems to assist end-users in achieving high performance when implementing grid applications. The models are analyzed using real infrastructure usage data. Different periods are investigated to assess the validity of the models under different workload conditions. Simplifications of the models are considered to lower their complexity. The findings of this paper are that tractable probabilistic models, taking into account faults and outliers but approximating the impact of the resubmission process, exhibit good performance and usability.

II. RELATED WORK

With the generalization of grid infrastructures exploitation for scientific production, large collections of grid usage traces become available. These traces are gaining a growing attention for the potential insight on grid systems they provide, and structured trace archive initiatives are emerging. They are used for statistical analysis of grid systems behavior. Probabilistic models have also been derived and

---

[1]EGEE European multi-disciplinary grid: http://www.eu-egee.org

exploited for addressing various optimization problems, especially related to fault-tolerance.

The problem of structured traces collection by itself is non-trivial. Grid traces are extracted out of the logs from many different and distributed middleware services. The logs information may be incomplete, and sometimes incoherent due to distributed resources synchronization problems. It may contain only partial information, or conversely produce very verbose low-level information that needs to be filtered out prior to analysis. Consequently, several groups are investing efforts in collecting usable grid usage trace sets. The Grid Workloads Archive [4] aims at collecting and organizing traces from different grid infrastructures. It also proposes data processing tools. The Real-Time Monitor (RTM) [5] and the Grid Observatory[2] (GO) are focusing on the EGEE production grid. The RTM gathers traces in near real-time for providing live usage information. It generates compact structured data out of the logs collected. The GO aims at collecting, structuring and archiving long-term traces for further analysis. The GO data is thus exploited in many computer science-related works [6], [9], [3], [8]. Both RTM and GO traces are exploited in this paper.

Beyond simple traces collection, many recent works have focused on post-analysis of the data archived [10], [11], [12]. In the AMon monitoring system [10], most relevant information on jobs submitted to EGEE is filtered out of the traces in order to help users to monitor experiments yielding large amounts of jobs. Due to the high failure rate characterizing grid infrastructures, many works are focusing on error detection and error cause identification. The GSTRAP system [9] aims at clustering traces from EGEE in order to detect and identify anomalies. Cieslak *et al* [11] are classifying errors using data mining techniques in order to help users to understand the reason for faults. Their study is based on CONDOR and experiments have been made both on a local grid and on the Open Science Grid[3]. Maier *et al* [12] pointed out the fact that error codes returned by systems do not always properly identify the real cause of failure. They are using data mining techniques on EGEE traces in order to determine the root cause for faults. Their methodology is decomposed into two steps: first building association rules and then pruning for deducing the restricted set of most relevant rules. In this paper, we are proposing two strategies for including faults and delays to failure in a probabilistic resubmission model but we did not investigated the causes of failure.

Statistics collected are further exploited for studying and modeling grid systems. Focusing on jobs management, Germain *et al* [6] have compared two user-level scheduling algorithms using recorded traces. Fault-tolerant scheduling methods have also been considered, such as rescheduling [13],

[14], [15] or short runs-based tests that apply to quite long, restartable jobs [16]. Ilijašić *et al* [3] have examined the relations between users, grid computing gateways and jobs. They have proposed probabilistic models in order to predict job abortion. Glatard *et al* also adopted probabilistic modeling approaches to estimate regular grid jobs latency [1] and more recently for studying pilot-jobs [17] on the EGEE production grid.

In a previous work we have modeled three different resubmission strategy [18]: the single resubmission, the multiple resubmission and the delayed resubmission strategies. However, faults where not considered in this work. In [8], we have demonstrated the necessity to take into account the latency in fault detection on a production grid, such a the EGEE. We have thus introduced the management of failures in the resubmission strategy proposed by [1] where each job is canceled and restarted if it has not started before a time-out value that is to be optimized. However, the model was based on a simplifying assumption that caused the time-out delay to be constant, ignoring potential faults recorded on the system. The purpose of this paper is to compare two implementations of the resubmission strategy: with or without adapting time-out dynamically. A proper model of strategy aim at optimizing the parameters of re-submission strategies that are needed to handle errors.

III. Grid jobs resubmission

A. Probabilistic modeling

The variable workload conditions and high failure rates encountered on grid infrastructures cause grid jobs to face non-negligible overheads. In the remainder, a job *latency* refers to the period of time between job submission and the start time of job execution on a grid computing resource. A fraction of *outliers*, never completing due to system faults resulting in complete loss of these jobs, is also observed. It is therefore compulsory for grid client applications to monitor the population of jobs submitted to the system and re-submit jobs which latency time is abnormally high to ensure completion of the computations. Determining the time-out threshold beyond which jobs need to be re-submitted is a non-trivial process as latencies are depending on the underlying grid infrastructure capability and workload.

To address this problem, a probabilistic model of the grid jobs latency is introduced below. Statistics on the grid infrastructure properties are collected through the analysis of grid usage traces to instantiate this model. An objective function of the jobs execution time expectation is then derived and optimized regarding to the time-out threshold. In the remainder, a capital letter X traditionally denotes a random variable with the probability density function (pdf) f_X and the cumulative density function (cdf) F_X. Jobs submitted can be either successful, faulty (due to a system error reported to the client) or outliers (lost without
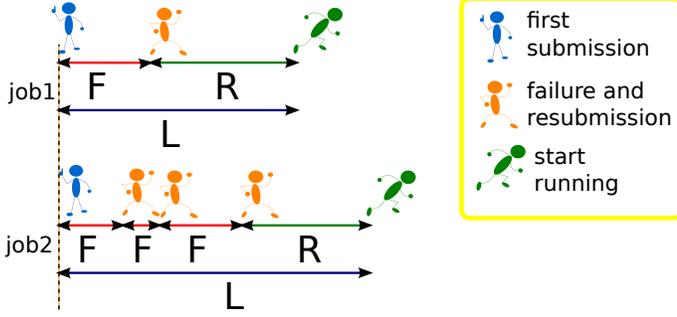
Figure 1. Two examples of job submission with resubmissions in case of failures.

any notification). We denote as ϕ the fraction of jobs encountering a system fault and ρ the fraction of outliers.

B. Model of the grid jobs latency

Let R denote the proper latency of a successful job and F denote the failure time of a faulty job. Assuming that faulty jobs are resubmitted without delay, let L denote the job latency taking into account the necessary resubmissions (see figure 1). L depends on the distribution of the jobs failure time. The probability, for a job to succeed is $(1 - \rho - \phi)$. In practice, the values of ρ, ϕ, and the distributions of R and F are numerically estimated from grid monitoring traces while L is computed as shown below.

A job encounters a latency $L < t$, t being fixed, if it is not an outlier and either:

- the job does not fail (probability $(1 - \rho - \phi)$) and its latency $R < t$ (probability $P(R < t) = F_R(t)$); or
- the job fails at $t_0 < t$ (probability $\phi f_F(t_0)$) and the job resubmitted encounters a latency $L < (t - t_0)$

The cumulative distribution of L is thus defined recursively by:

$$F_L(t) = (1 - \rho - \phi)F_R(t) + \phi \int_0^t f_F(t_0).F_L(t - t_0)dt_0$$

As shown in a previous work [8], the cdf F_L can be numerically estimated by discretizing this equation (introducing the second as the discretization step for the variable t) and observing that no successful job has a null latency ($F_R(0) = 0$)[4]. This results in the following recursive expression of F_L:

$$
\begin{aligned}
F_L(0) &= 0 \\
F_L(1) &= \frac{1 - \rho - \phi}{1 - \phi f_F(0)} F_R(1) \\
F_L(t > 1) &= \frac{1}{1 - \phi f_F(0)} \Bigg[(1 - \rho - \phi) F_R(t) \\
&\quad + \phi \sum_{u=1}^{t-1} f_F(t - u) F_L(u) \Bigg]
\end{aligned}
$$

(1)

[4]For a discussion on the validity of these hypotheses, refer to [8]

Under the hypothesis that failed jobs reporting requires at least one second (and therefore $f_F(0) = 0$), this equation simplifies:

$$
\begin{aligned}
F_L(0) &= 0 \\
F_L(1) &= (1 - \rho - \phi)F_R(1) \\
F_L(t > 1) &= (1 - \rho - \phi)F_R(t) \\
&\quad + \phi \sum_{u=1}^{t-1} f_F(t - u) F_L(u)
\end{aligned}
$$

(2)

This hypothesis is valid in the sense that it has negligible impact on the jobs expected latency optimization procedure as will be shown later.

The probabilistic law L approximated through the discrete equation 2 can be exploited to numerically estimate the expected latency of jobs submitted to the grid infrastructure. This expectation depends on L and the resubmission strategy adopted as described in the following sub-sections. To be of practical interest, the derived expectation model needs to remain mathematically and computationally tractable.

C. Resubmission strategy and time-out estimation

Various client-side re-submission and multi-submission strategies have been considered in the literature to improve grid performance [7], [18]. In this paper we focus on the simplest and most common simple resubmission strategy, where a job is canceled and resubmitted if its latency is higher than a time-out value t_∞. Poor estimations of the time-out values can have strong performance impact: a lower time-out value will cause potentially successful jobs to be cancelled too early, while a higher time-out value will cause penalizing delays for non-successful jobs. However, the value of t_∞ can be determined through an optimization procedure [1]. Let J denote the total latency experienced by a job, including as many resubmissions as needed after the time-out threshold t_∞ has been reached.

A simplified model of the expected latency taking into account resubmissions as a function of the time-out ($E_J(t_\infty)$) is proposed in [1]. However, this model excludes faults (faulty jobs were excluded from the statistics collection). To include faults in the resubmission process, we describe below two alternatives to this model (see figure 2):

J_0 Jobs whom proper latency R is greater than the time-out value t_∞ are canceled and resubmitted. The resulting total latency is denoted J_0.

J_1 Jobs for which the latency L, including resubmissions due to faults, is greater than the time-out value t_∞ are canceled and resubmitted. The resulting total latency is denoted J_1.

Strategy J_1 is an approximation of strategy J_0 when faults occur: it does not take into account the time spent before fault notification in the resubmission delay. The time-out threshold will therefore be reduced by this period of time. However, the model of J_1 is simpler and computationally
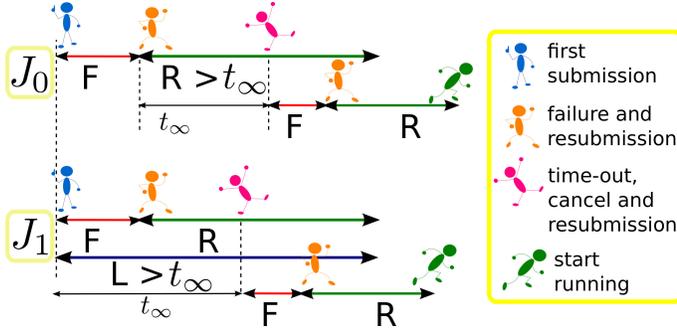
Figure 2. Two ways of taking faults into account in the resubmission strategy J_0 (with time-out adapted dynamically in case of failures) and J_1 (with fixed time-out, independently of failures).

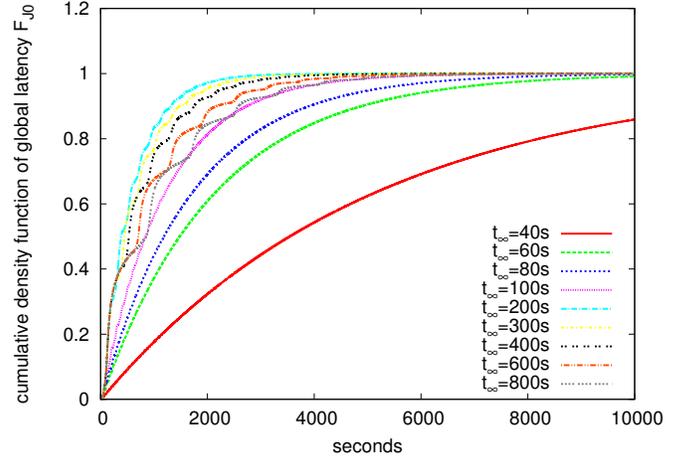

Figure 3. Strategy J_0. Cumulative density functions of global latency including resubmission for different time-out values (t_∞).

more efficient. In [8], only strategy J_1 was introduced. In this paper, we also describe J_0 and show how it can be approximated by J_1.

D. Strategy J_0

Strategy J_0 cancels and restarts a job only when its proper latency is greater than the time-out value t_∞. Since no job has a null latency R, we have:

$$f_{J_0}(0) = 0$$

Before the time-out value t_∞, no job has been canceled. There are two possibilities for a job to have a total latency of t with $0 < t < t_\infty$:

- no failure and latency equal to t: $(1 - \rho - \phi)f_R(t)$
- at least one failure at t_0: $\phi f_F(t_0)f_{J_0}(t - t_0)$

leading to:

$$f_{J_0}(t) = (1 - \rho - \phi)f_R(t) + \phi \sum_{t_0=1}^{t-1} f_F(t_0)f_{J_0}(t - t_0)$$

After the time-out value t_∞, the job has already at least failed once or time-outed once. A time-out occurs in case of:

- outlier, with probability: ρ
- latency to failure greater than t_∞, with probability: $\phi(1 - F_F(t_\infty))$
- latency R greater than t_∞, with probability: $(1 - \rho - \phi)(1 - F_R(t_\infty))$

leading to:

$$
\begin{aligned}
f_{J_0}(t) &= (\rho + (1 - \rho - \phi)(1 - F_R(t_\infty)) \\
&\quad + \phi(1 - F_F(t_\infty)))f_{J_0}(t - t_\infty) \\
&\quad + \phi \sum_{t_0=1}^{t_\infty} f_F(t_0)f_{J_0}(t - t_0)
\end{aligned}
$$

The complete expression of f_{J_0} is given by equation 3:

$$
\begin{aligned}
f_{J_0}(0) &= 0 \\
f_{J_0}(0 < t < t_\infty) &= (1 - \rho - \phi)f_R(t) \\
&\quad + \phi \sum_{t_0=1}^{t-1} f_F(t_0)f_{J_0}(t - t_0) \\
f_{J_0}(t \geq t_\infty) &= (\rho + (1 - \rho - \phi)(1 - F_R(t_\infty)) \\
&\quad + \phi(1 - F_F(t_\infty)))f_{J_0}(t - t_\infty) \\
&\quad + \phi \sum_{t_0=1}^{t_\infty} f_F(t_0)f_{J_0}(t - t_0)
\end{aligned}
\tag{3}
$$

Figure 3 shows the profiles of the cdf total latency including resubmissions with strategy J_0 corresponding to different time-out values. The curve corresponding to $t_\infty = 200$ seconds is the closest to the optimal found among this sample: it displays the fastest convergence towards 1.

From the expression of f_{J_0} in equation 3, the latency expectation computation is straight forward using its mathematical definition:

$$E_{J_0}(t_\infty) = \int_0^\infty t f_{J_0}(t)dt \tag{4}$$

A typical profile of the expectation of the total latency, as observed on a production grid, including all resubmissions and computed from equation 4, is plotted in figure 4. In this case, the curve reaches a minimum value $E_{J_0} = 583s$ for an optimal time-out value $t_\infty = 191s$. An underestimation of t_∞ would cause early jobs cancellation, thus increasing the number of resubmission and the total latency J_0. Conversely, an overestimation of t_∞ would penalize the non-successful jobs by late resubmission. This result is coherent with the observation from figure 3 where the best curve profile is obtained with t_∞=200s.

E. Strategy J_1

In strategy J_1, in case of failure the job time-out is not increased by the delay expired before failure reporting, thus under-estimating the time-out of the resubmitted job. Under this simplification hypothesis, the distribution of job latency including resubmissions f_L can be used to derive the distribution f_J in case of failure as observed in [8]. The latency J_1 is therefore defined as a function of L and t_∞: a job for which L is greater than t_∞ is cancelled and resubmitted, thus increasing J_1 by the time-out delay. Observing that $F_L(t)$ corresponds to the probability for a job to succeed with a latency lower than t, the probability for a job to time-out is $q = P(L > t_\infty) = 1 - F_L(t_\infty)$. Denoting n the number of times the job timed-out (n is the integer such that $t \in [nt_\infty, (n+1)t_\infty]$):

$$
\begin{aligned}
F_{J_1}(t) &= P(L < nt_\infty) \\
&\quad + P(nt_\infty < L < t \mid t \le (n+1)t_\infty) \\
&= 1 - q^n + q^n F_L(t - nt_\infty)
\end{aligned}
$$

(See [1] for details). Consequently, $f_{J_1}(t) = q^n f_L(t - nt_\infty)$ and the expectation of J_1 is $E_{J_1}(t_\infty) = \int_0^\infty u f_{J_1}(u)du$:

$$
E_{J_1}(t_\infty) = \frac{1}{F_L(t_\infty)} \int_0^{t_\infty} (1 - F_L(u))du \tag{5}
$$

Minimizing this expression of E_{J_1} yields the optimal time-out value t_∞=195s for a minimal value $E_{J_1} = 586s$. The profile of E_{J_1} is plotted on figure 4.

Note that the model of latency J_1 was derived from equation 2, assuming that $f_F(0) = 0$. To validate this hypothesis, the result obtained by this model can be compared to the one obtained by a model derived from equation 1 [8]. The impact on the estimated execution time is lower than 0.005%, confirming the hypothesis.

It should be noted that besides its more complex mathematical representation, model J_0 is significantly more compute intensive than model J_1. If we denote by $t_{\infty MAX}$ the upper bound of the interval of search for optimal timeout and n the number of samples needed for the computation of 4, the complexity of the computation of E_J in case of J_0 is in $nt_{\infty MAX}^2$ while it reduces to $t_{\infty MAX}^2$ in case of J_1. Numerically, we took $t_{\infty MAX} = 1000$ and $n = 100000$.

IV. QUANTIFYING THE IMPACT OF FAULTS ON RESUBMISSION STRATEGIES

The results obtained by exploiting strategies J_0 and J_1 are compared to assess the performance of the simplified model in place of the exact one. Different experimental settings are considered to validate the results over infrastructures exhibiting different properties and under variable workload conditions along time. The first experiment targets realistic conditions as observed on the EGEE production grid infrastructure. The second experiment targets different infrastructures by varying the model parameters (ratios of faults,
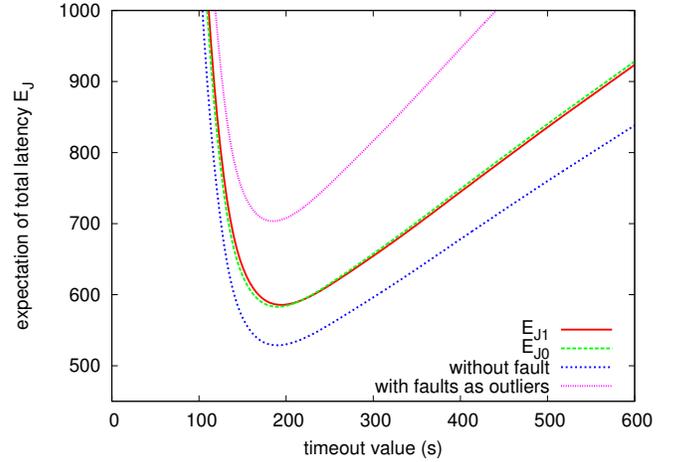

Figure 4. Expectation of total latency with respect to the time-out value t_∞: cases of strategy J_1, strategy J_0, a simple model ignoring failures and an approximation where failures are accounted as outliers.

outliers and distribution). The third and fourth experiments target workload variability as observed along time on the EGEE grid.

A. Impact on the EGEE production grid infrastructure

The data considered in this study are RTM/Grid Observatory traces of the EGEE grid activity during the period from September 2005 to June 2007. 33,419,946 job entries were collected, each of them representing a complete job run. Among these data, 64.8% corresponds to successful jobs, 19.1% to failures and the remaining 16.1% to outliers. The distribution of latency of successful jobs (R) and failure time of a faulty job (F) are computed from these data.

Figure 4 displays several plots of the expectation of the total latency including resubmission. We observe that the results for strategies J_0 and J_1 are very close. In order to compare the difference between J_0 and J_1 with other hypotheses, E_J is also plotted for two other cases. The case "without fault" where data corresponding to faulty jobs are neglected leads to an underestimation of E_J while the case "with fault as outliers" where faulty jobs are considered as outliers leads to an overestimation of E_J.

This experiment shows that taking into account the latency for faults detection has a higher impact on the parameters estimation than the model used (J_0 or J_1). Moreover, for J_1, the best time-out value is $t_\infty = 195s$ for an optimal $E_{J_1} = 586s$ while, for J_0, the best time-out value is $t_\infty = 191s$ for an optimal $E_{J_0} = 583s$. If we would have chosen the time-out value from J_1 to be applied with strategy J_0, we would have obtained $E_{J_0}(195) = 583s$ (valued rounded to integer value). The relative difference with the optimal value is negligible, in the order of 0.06%.

Looking at the distribution of faults detection latency f_F (figure 5), we observe that a pick is centered at 7 seconds
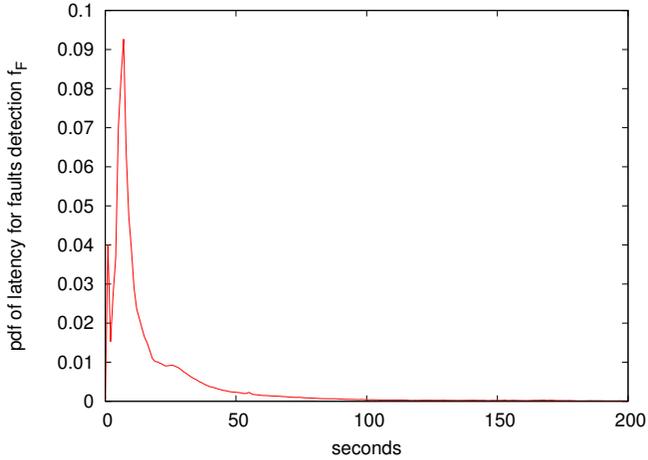
Figure 5. Probability density function of the latency for fault detection. We observe a maximum at 7 seconds.
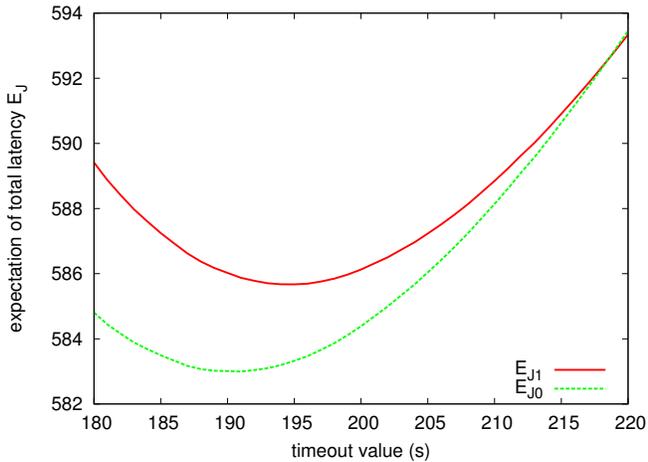

Figure 7. Varying the ratio of faults ϕ in case of J_0. The optimal time-out value does not vary while the expectations of total latency are increasing.

ϕ	t_{∞_0}	$E_{J_0}(t_{\infty_0})$	t_{∞_1}	$E_{J_0}(t_{\infty_1})$	$\Delta\%$
19%	191	583	195	583	0.06
21%	191	590	195	590	0.08
25%	191	606	196	607	0.08
30%	191	630	198	631	0.14
40%	191	694	202	696	0.32
50%	191	795	207	780	0.61

Table I
COMPARISON OF STRATEGIES J_0 AND J_1 WHEN VARYING THE RATIO OF FAULTS ϕ.

Figure 6. Zoom on the expectation of total latency for strategies J_0 and J_1.

with 50% of failures detected in no more than 10 seconds. On figure 6, we observe that variations of 10 seconds of time-out value around the optimal value does not increase the expectation of execution time by more than 2 seconds (0.3%). Consequently, J_0 and J_1 results are very close.

B. Impact on different workloads

J_0 and J_1 results are hardly differentiable under the conditions observed on the EGEE production infrastructure during the period 2005-2007. Different workloads are simulated by artificially varying the model parameters (failure distribution parameterization) in order to determine under which conditions the two strategies produce different results.

1) Varying the fault ratio ϕ: An increasing ratio of faults ϕ is considered, ranging from the value measured on EGEE (19%) to 50% (see figure 7). They all conduct to an optimal
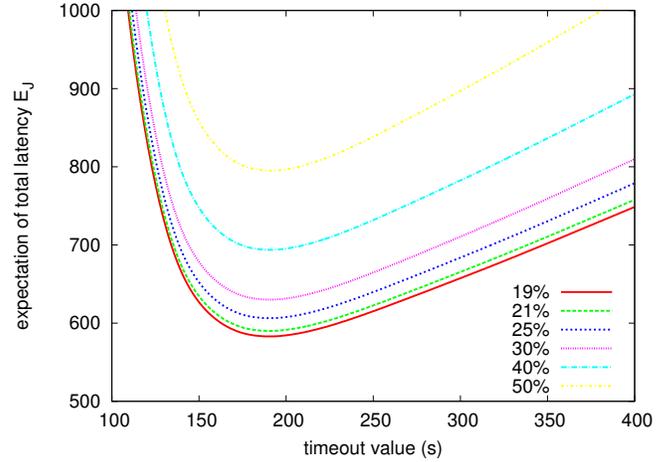
time-out value $t_{\infty_0} = 191s$ in the case of J_0. For each ϕ value, we have computed the optimal time-out in the case of J_1, t_{∞_1}, and compared the optimal value E_{J_0} with the one obtained at t_{∞_1}. Results are reported in table I. Even with the largest value of ϕ, the relative difference does not exceed 0.6%.

2) Translating the pdf of failure detection latency: As another step toward worse experimental conditions, the latency of the faults detection delay was increased up to 1000 seconds (see figure 8).

For each delay of fault detection, the optimal time-outs t_{∞_0} and t_{∞_1} were computed with strategies J_0 and J_1, and the expectation E_{J_0} was estimated using both values. Results are reported in table II. The relative difference grows up to 1.3% for an increased delay by approximately 90 seconds. For higher delays, relative differences are decreasing until 200 seconds where they do not vary any more: all faulty jobs encounter a time-out.

For delay increases smaller than 50 seconds, the relative difference is less than 0.6%.

From these artificial varying conditions on faults, we can conclude that, even with variations of faults ratio up to 50% and increase of pdf of latency for faults detection up to 50 seconds, numerically, results given by strategies J_0 and J_1
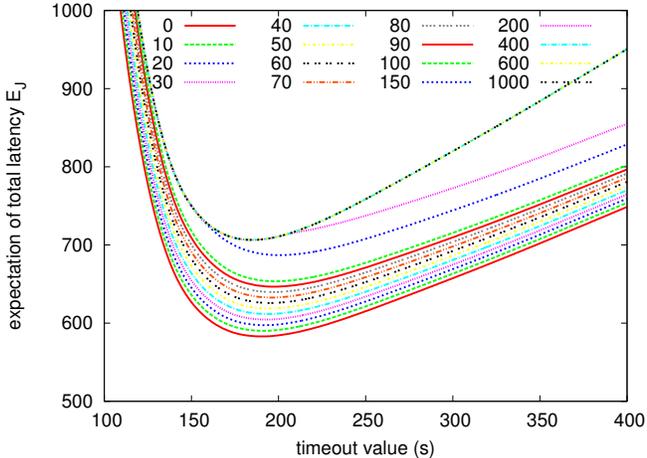
Figure 8. Translating pdf of failure detection latency: impact on E_{J_0}. Both optimal time-out value and expectation of total latency are varying. For translation higher than 200 seconds, results are similar since the optimal time-out is lower than 200 seconds.

t	t_{∞_1}	t_{∞_0}	$E_{J_0}(t_{\infty_0})$	$E_{J_0}(t_{\infty_1})$	$\Delta\%$
0s	195s	191s	583.3s	583s	0.06
10s	198s	191s	591s	590s	0.1%
20s	202s	192s	599s	597s	0.3%
30s	206s	193s	607s	605s	0.4%
40s	208s	194s	614s	612s	0.5%
50s	211s	194s	622s	619s	0.6%
60s	214s	196s	630s	626s	0.7%
70s	219s	196s	639s	633s	1.0%
80s	223s	196s	648s	640s	1.2%
90s	225s	198s	655s	647s	1.3%
100s	218s	198s	690s	687s	0.5%
150s	187s	200s	690s	687s	0.4%
200s	185s	185s	707s	707s	0%
>200s	185s	185s	707s	707s	0%

Table II
COMPARISON OF STRATEGIES J_0 AND J_1 WITH INCREASED FAILURE DETECTION LATENCY.

are very close.

C. Test on 2010 data

Recent data on the EGEE production infrastructure available from the Grid Observatory (period from 2010-03-29 to 2010-04-04) was tested. After curation, 394315 data entries are suitable for computations. A classification of all entries, similar to [8] but adapted to new fault conditions reported (due to evolution of the middleware), was performed.

Compared to the data used in this paper, a lot of failures are reported very quickly, thus changing the profile of E_J (see figure 9). However, E_J still exhibits a global minimum.

With a ratio of failure of $\phi = 25.8$ % and a ratio of outliers of $\rho = 9.4$ %, the optimal time-out value t_∞ is of 349 seconds for both J_0 and J_1: there is no added benefit to consider strategy J_0 instead of J_1.
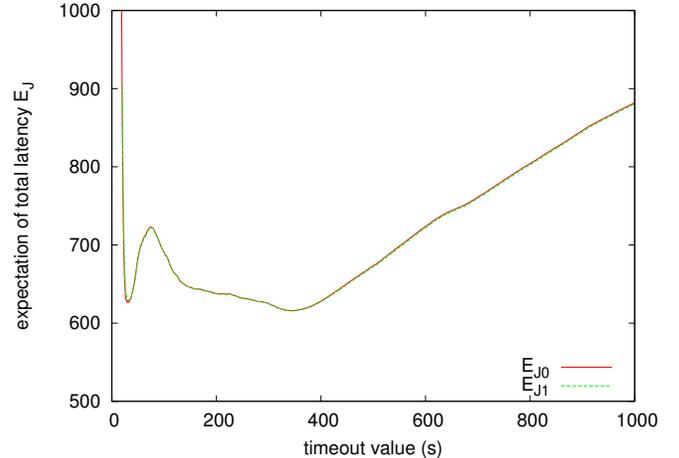

Figure 9. 2010 data. Expectation of execution time including resubmissions for strategies J_0 and J_1: results are very similar even if the profile differs from the one obtained with older data (see for example figure 4).

D. Test on 2009 data

Further tests were conducted using Grid Observatory / RTM data available earlier in 2009/2010: weeks 28 to 37 in 2009, week 14 in 2010. The results are identical for all weeks except the week 31 in 2009 where there is a relative difference of 0.02% between the expectation of total latency from the optimal values computed for strategies J_0 and J_1. The profiles of E_J are similar to the one presented in figure 9.

V. CONCLUSION

Production grids usage is hampered by high failures rate and highly variable latencies observed in large scale complex systems. Efficient grid jobs resubmission therefore becomes a key feature of any grid experiment production environment. However, the scale of contemporary grids and their non stationary workloads makes optimal resubmission parameterization difficult.

In this paper, two probabilistic models of the simple grid jobs resubmission strategy were introduced. In case of failures, strategy J_0 is properly taking into account the time spent before failure notification in the estimation of the time-out threshold beyond which jobs are canceled and resubmitted. Strategy J_1 is an approximation neglecting this notification time.

Experiments with J_0 and J_1 using the same sets of data show very close results in terms of expected latency time under variable workload conditions. This result has been validated both on real EGEE trace data at different times and on infrastructures with different faults distributions. The model can easily be adapted to other production environments by measuring few infrastructure-specific parameters (*i.e.* ρ, ϕ, R, F).

Strategy J_0 is computationally more complex than J_1, that can be used as a valid approximation in practical implementations. This result can be extended and under similar conditions, we could benefit from this simplified model to address more elaborated client-side resubmission strategies such as multiple submission [7] and delayed resubmission with overlap of multiple instances of submitted jobs [18], that are exploited by production grid users today.

REFERENCES

[1] T. Glatard, J. Montagnat, and X. Pennec, "Optimizing jobs timeouts on clusters and production grids," in *International Symposium on Cluster Computing and the Grid(CCGrid'07)*. IEEE, May 2007, pp. 100–107. [Online]. Available: http://rainbow.polytech.unice.fr/publis/glatard-montagnat-etal:2007a.pdf

[2] E. Medernach, "Workload Analysis of a Cluster in a Grid Environment," in *Job Scheduling Strategies for Parallel Processing(JSSPP)*, Jun. 2005, pp. 36–61. [Online]. Available: http://www.cs.huji.ac.il/~feit/parsched/jsspp05/p-05-2.pdf

[3] L. Ilijašić and L. Saitta, "Statistical Characterization of a Computer Grid," in *International Symposium on Methodologies for Intelligent Systems(ISMIS)*, vol. LNAI 5722. Springer Verlag, Sep. 2009, pp. 523–532.

[4] A. Iosup, H. Li, M. Jan, S. Anoep, C. Dumitrescu, L. Wolters, and D. Epema, "The Grid Workloads Archive," *Future Generation Computer Systems (FGCS)*, vol. 24, no. 7, pp. 672–686, Jul. 2008.

[5] D. Colling, J. Martyniak, S. McGough, A. Křenek, J. Sitera, M. Mulač, and F. Dvořák, "Real Time Monitor of Grid job executions," in *Computing in High Energy Physics / Journal of Physics: Conference Series(CHEP 2009)*. IOP Publishing, Mar. 2009. [Online]. Available: http://www.iop.org/EJ/conf

[6] C. Germain, C. Loomis, J. T. Mościcki, and R. Texier, "Scheduling for Responsive Grids," *Journal of Grid Computing (JOGC)*, vol. 6, no. 1, pp. 15–27, Mar. 2008.

[7] H. Casanova, "On the Harmfulness of Redundant Batch Requests," in *15th IEEE International Symposium on High Performance Distributed Computing(HPDC'06)*, Jun. 2006, pp. 255–266.

[8] D. Lingrand, J. Montagnat, J. Martyniak, and D. Colling, "Optimization of jobs submission on the EGEE production grid: modeling faults using workload," *Journal of Grid Computing (JOGC) Special issue on EGEE*, vol. 8, no. 2, pp. 305–321, Mar. 2010.

[9] X. Zhang, M. Sebag, and C. Germain, "Multi-scale Real-time Grid Monitoring with Job Stream Mining," in *9th IEEE/ACM International Symposium on Cluster Computing and the Grid(CCGrid)*, May 2009, pp. 420–427. [Online]. Available: http://www.lri.fr/~xlzhang/papers/ccgrid09_zhang.pdf

[10] H. Eichenhardt, R. Müller-Pfefferkorn, R. Neumann, and T. William, "User- and job-centric monitoring: Analysing and presenting large amounts of monitoring data," in *9th International Conference on Grid Computing(IEEE Grid)*. IEEE, Sep. 2008, pp. 225–232.

[11] D. Cieslak, N. Chawla, and D. Thain, "Troubleshooting thousands of jobs on production grids using data mining techniques," in *9th International Conference on Grid Computing(IEEE Grid)*. IEEE, Sep. 2008, pp. 217–224. [Online]. Available: http://www.nd.edu/~ccl/research/pubs/debug-grid08.pdf

[12] G. Maier, D. C. Vanderster, and D. Kranzlmueller, "Finding associations in Grid monitoring data," in *10th International Conference on Grid Computing(IEEE Grid)*. IEEE, Oct. 2009, pp. 89–96. [Online]. Available: http://maierg.web.cern.ch/maierg/documents/grid09_paper.pdf

[13] C. Dabrowski, "Reliability in grid computing systems," *Concurrency and Computation: Practice & Experience (CCPE) Special issue on Open Grid Forum*, vol. 21, no. 8, pp. 927–959, Jun. 2009. [Online]. Available: http://www.nist.gov/itl/antd/upload/Dabrowski-GridReliabilityEarlyView.pdf

[14] E. Frachtenberg and U. Schwiegelshohn, "New Challenges of Parallel Job Scheduling," in *13th Job Scheduling Strategies for Parallel Processing(JSSPP)*, vol. LNCS 4942. Springer, Apr. 2008, pp. 1–23. [Online]. Available: http://www.it.irf.uni-dortmund.de/IT/Publikationen/pdf/FS08.pdf

[15] E. Huedo, R. S. Montero, and I. M. Llorente, "Evaluating the reliability of computational grids from the end user's point of view," *Journal of Systems Architecture*, vol. 52, no. 12, pp. 727–736, Dec. 2006.

[16] O. Thebe, D. P. Bunde, and V. J. Leung, "Scheduling Restartable Jobs with Short Test Runs," in *14th Workshop on Job Scheduling Strategies for Parallel Processing(JSSPP'09), workshop: IPDPS*, vol. LNCS 5798. Springer, May 2009, pp. 116–137.

[17] T. Glatard and S. Camarasu-Pop, "Modelling pilot-job applications on production grids," in *7th international workshop on Algorithms, Models and Tools for Parallel Computing on Heterogeneous Platforms(Heteropar'09)*, Aug. 2009.

[18] D. Lingrand, J. Montagnat, and T. Glatard, "Modeling user submission strategies on production grids," in *International Symposium on High Performance Distributed Computing(HPDC'09)*, Jun. 2009, pp. 121–130. [Online]. Available: http://rainbow.polytech.unice.fr/publis/lingrand-montagnat-etal:2009.pdf