Comments on the revised version

March 9, 2018

Let us first sincerely thank the reviewers for their relevant and in-depth remarks once again. Their comments helped to further discuss the results, and to find a typographical error in the presentation of the results that passed our scrutiny.

We detail our answers to rewiewer's comments (bold).

1 Reviewer 1

I basically liked this paper and support its publication. I have only very few and minor comments.

In sect 1.1 you claim 2 "new" online approaches, and in 1.2 you call them "original". Your work is valid even if you admit that both have been suggested before.

Last paragraph of 1.2 is the third repetition of this. It is redundant, despite the fact that some reviewer didn't get it.

You are right. We rephrased this in a more neutral way.

In sect 5.1 you explain that you set lambda=1 because due to the resampling the distributions are stationary. That is correct. But it would have been good to also conduct simulations on the original traces, which are probably not stationary, and therefore there is more to gain from switching between policies. Also, instead of starting the simulation of each period from an empty system, why not start from the situation at the end of the previous period?

Regarding your first point, this is a good remark, we based this experimental design around previous results from "Tuning Backfilling Queues"(JSSPP17) that show some degree of stationarity using a train/test procedure. We clarified this in the text. We indeed expect that non-stationary policy adaptation may allow more improvements (while simultaneously being much harder to achieve).

Your second point is also relevant: starting the simulation at the end of the previous period should improve the results. But we can see that the 'full feedback' simulated policy here is almost as good as the best policy in the set, so we did not worry about this. In a non-stationary case, it may indeed be necessary to properly take care of this. We added a remark in section 5.1:

> The above cost directly corresponds to the cumulative waiting time of the policy over the preceding periods (more precisely to the cumulative simulated estimate of the waiting time of the policy over the preceding periods) so that the length of the period has little impact here. The only bias comes from the boundary states of the simulation. Indeed, we run simulations from an empty system and wait for the system to be empty when job submissions cease at the end of the period (see Figure **??**). Note that starting the simulation using the system state at the end of the previous period should improve the results. Here it will not be necessary hence we keep the simple version with an empty system. However, it may prove useful in the non-stationary case mentionned above.

Typo at end of 6.1.3: sentence with [24] appears twice.

Right, we fixed this.

2 Reviewer 2

In comparison to the first version, the authors have made several small changes that results in minor improvements of the manuscript. Nevertheless, the main problems of the manuscript remain. Therefore, the position of the reviewer remains unchanged.

The authors present a study based on a selection of several strategies and parameters. They execute several simulations and show that using a feedback mechanism improvements over the standard approach without feedback are possible. This results is not surprising considering previous results some of which are referenced within the manuscript. It must be noted that the authors do not suggest a specific new scheduling algorithm and show that this algorithm results in significant improvements over existing methods.

We regret that the detailed answer to your comments did not convince you. We disagree with this statement. We indeed provide a new scheduling algorithm, which adapts to the situation by selecting the correct reordering policy. This algorithm results in significant improvements over existing methods.

We thank you once again for your in-depth comments which we integrated into the current revision.

3 Reviewer 3

I think that the simulation model presented in this paper is carefully created and it shows convincing results. However, there are two questions about the discussion for the simulation results:

The results in Table 3 and Table 4 show that the performance of the static strategy, LQF, is comparable with the presented online tuning strategies, Full and Bandit. Which strategy is better in the real world? We can say that LQF may be preferable because they are less complicated and run with smaller scheduling overhead. The authors should add the discussion to compare LQF and the online tuning strategies.

First of all, we corrected a crucial typographical error in the text that slipped our scrutiny at the time of running the build chain for the publication. The S and L were inverted in both Table 3 and Figure 4. This resolves a blatant consistency issue with other publications, most crucially "Tuning Backfilling Queues"(JSSPP17).

You are right, the discussion you suggest is warranted. We added it to paragraph **SQF versus online tuning** in Section 6.3.2:

> Table 3 shows that the SQF policy works quite well on the traces considered here on average, and under this resampling scheme. However, the variability is high, and this policy is not the best on all traces. It interesting to have an experimental evaluation of exactly how much one gains on average by using an online policy versus a leave-one-trace-out majority vote policy choice. However, a good estimation of this quantity is tricky. Indeed, we expect the change in the relative performance of policies to increase with two factors. First, simply adding workload traces to our selection will naturally show more cases with a different behavior. Second, factoring in the topology model of the machines should increase the heterogeneity between platforms and therefore increase the dependence of the simulation performance on the trace/platform. Such a study would be more ambitious and is beyond the scope of this study. Here the need for adaptivity is actually understated by the simulation approach.

The authors conclude that simulation-based strategy is best but the bandit-based strategy is easier to use and cheaper to run. How easy and how cheaper can we run the bandit-based strategy? The authors should add the quantitative discussion how the bandit-based strategy is easy to use and saves cost (or scheduling overhead) compared with the simulation-based strategies.

Right, this point needs a clarification. We added the following discussion to paragraph **SQF versus online tuning** in Section 6.3.2:

> While we can not conclude on how much one gains by not using SQF in this case, we can comment on the added complexity and computing overhead of using the simulation-based and bandit online strategies. In both cases, there is no overhead in scheduling decisions as the policy is fixed during the period (one day, or week). The bandit strategy has a negligible computational overhead at the end of the period. Its two computational operations are maintaining a sequential average and generating a pseudo-random number. The simulation-based strategy requires to simulate the system K times every period end, where K is the number of alternative policies considered (here, K=12). Using our simulator, this takes a few seconds. Using a more precise simulator with topological modeling, one can argue that the overhead is still manageable since these simulations can be easily done in parallel. The most important constraint is that these simulations should not take too much time compared to the period size, in order not to delay too much the policy switch decision.