Supplementary file for paper "Competition-based failure-aware scheduling for high-throughput computing systems on peer-to-peer networks"

Carlos Pérez-Miguel Alexander Mendiburu
Jose Miguel-Alonso
Intelligent Systems Group,
Department of Computer Architecture and Technology,
School of Computer Science.
University of the Basque Country UPV/EHU.
{carlos.perezm, alexander.mendiburu, j.miguel}@ehu.es

February 4, 2015

When modeling the on-line and off-line times of nodes in a distributed system, two main approaches can be found in the literature. Often it is assumed that the failure/reparation events can be represented using exponential distributions, but Weibull distributions are also commonly used. In the main body of paper "Competition-based failure-aware scheduling for high-throughput computing systems on peer-to-peer networks" our choice was to use exponential distributions. In this supplementary file we repeat the experiments carried out in the main paper, but using Weibull distributions to control the failure and reparation behavior of the (simulated) nodes. These additional experiments show that the advantages of our competition-based failure-aware scheduling proposal is not tied to the distribution used to model nodes' behavior.

A Weibull distribution has two parameters: *scale* and *shape*. Regarding the shape parameter, Javadi et al. analyzed in [1] several failure traces from real systems and concluded that, when describing the failure distribution of a certain node, this parameter takes values in the range $(0.33 - 0.85)$, which implies that the failure rate decreases over time. Thus, we have decided to fix this parameter to 0.7 for both, stable and unstable, types of nodes. With respect to the shape parameter of the repair distribution, in the same study they concluded that it takes values in the range $(0.35 - 0.65)$, so we have fixed this parameter to 0.5 for both types of nodes.

As regards to the scale parameters of the Weibull distributions used, we have followed this approach to select them. Given a target, expected on-line, or off-line, time $E[X]$ for a certain node, the λ parameter, *rate*, of an exponential with that mean is simply:

$$E[X] = \frac{1}{\lambda} \tag{1}$$

However, for the Weibull distribution with parameters λ (scale) and k (shape), the expression is:

$$E[X] = \lambda\Gamma\left(1 + \frac{1}{k}\right) \tag{2}$$

Consequently, as the shapes have been already fixed, we can compute the scale of a Weibull distribution with the target expected mean as:

$$\lambda = \frac{E[X]}{\Gamma\left(1 + \frac{1}{k}\right)} \tag{3}$$

Using this equation, and in order to achieve the same target failure and reparation times used in the experiments with exponential distributions, the parameters selected to model the nodes using Weibull distributions are:

- Stable nodes:
  - Failures:
    - $scale = 789999.5$.
    - $shape = 0.7$.
  - Reparations:
    - $scale = 5000$.
    - $shape = 0.5$.
- Unstable nodes:
  - Failures:
    - $scale = 7899.995$.
    - $shape = 0.7$.
  - Reparations:
    - $scale = 500$.
    - $shape = 0.5$.

The remaining parameters used to run the experiment are those described in Section 6,"Experimental environment", of the main paper.

The results of the experiments comparing different scheduling algorithms, including distributed and failure-aware ones, are summarized in Figures 1, 2, 3 and 4. As can be seen, results are very similar to those presented in the main paper, although the differences between algorithms are now narrower. In some cases, FR is slightly better (0.2%) than our proposals in terms of make-span. However, note that FR is a centralized algorithm with lower overheads than our distributed proposals. Therefore, this set of experiments confirm the good behavior of our algorithms. In particular, BFGC is globally the best of the tested options.

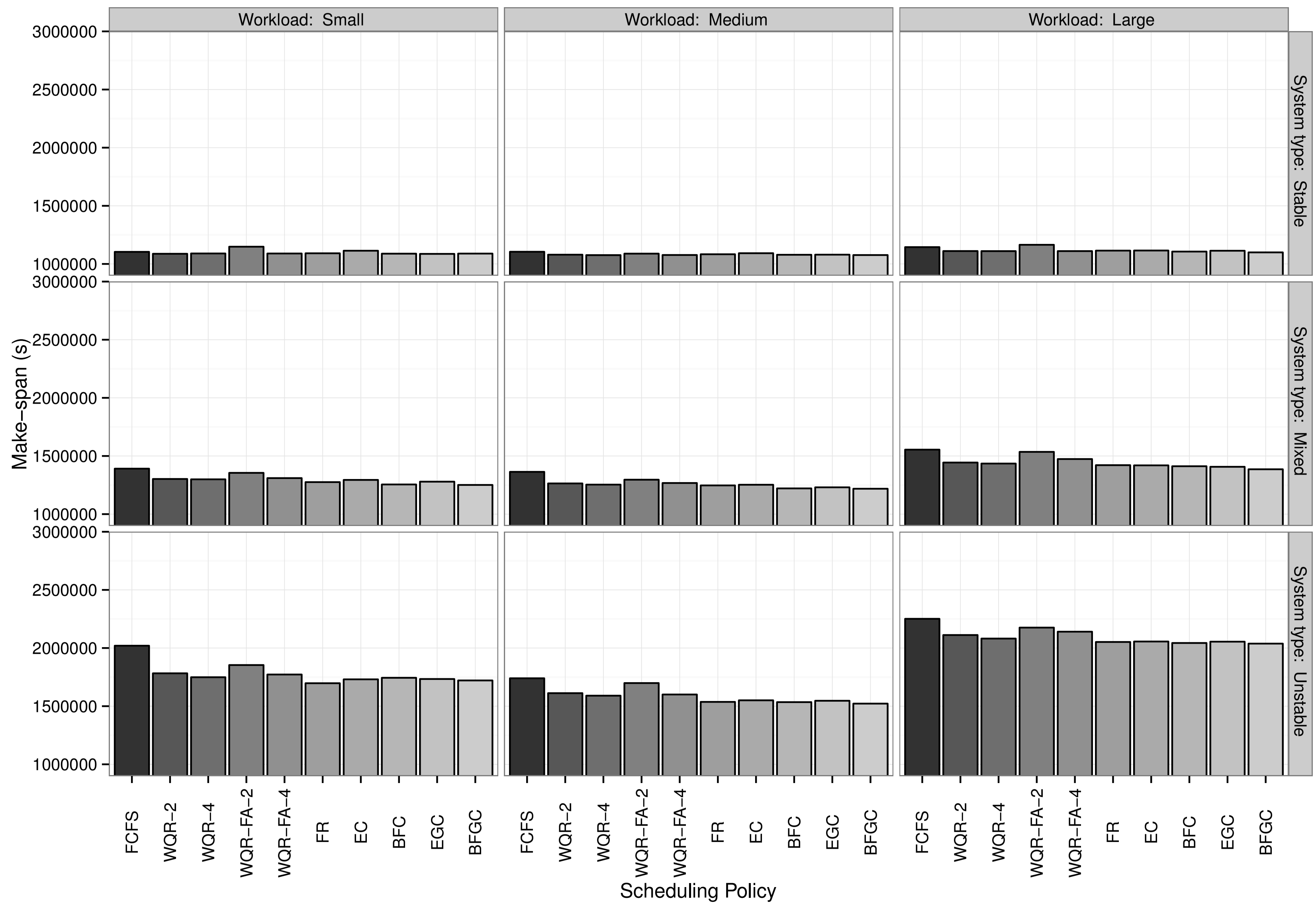


Figure 1: Make-span using different scheduling policies for different scenarios (combinations of node stability and task size). The ideal make-span is 1000000.

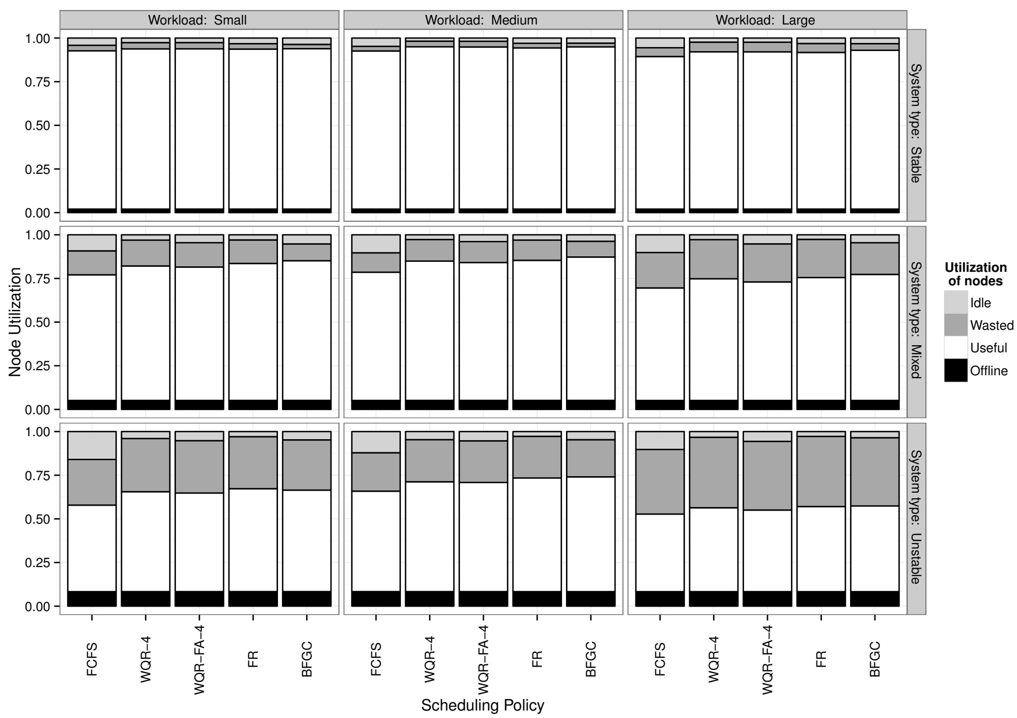
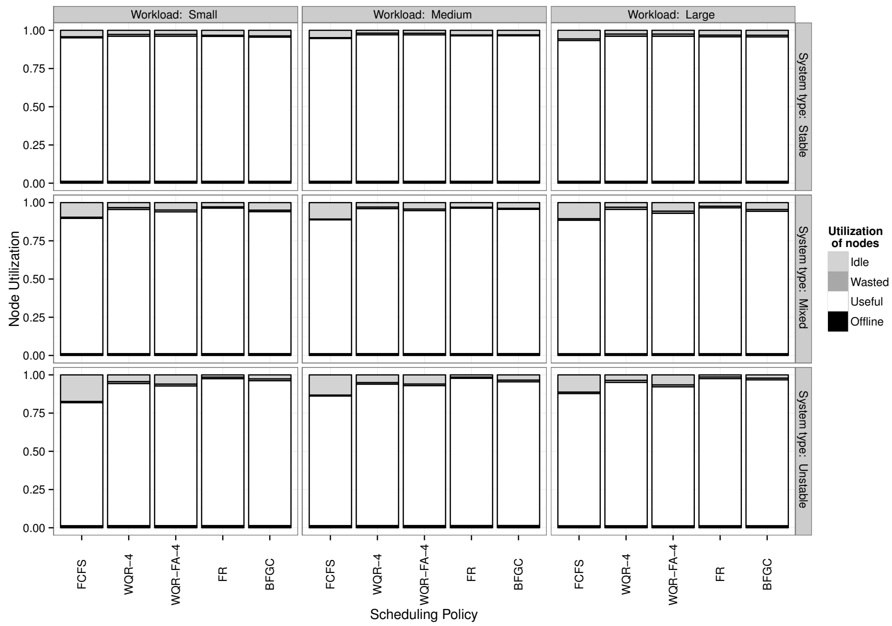
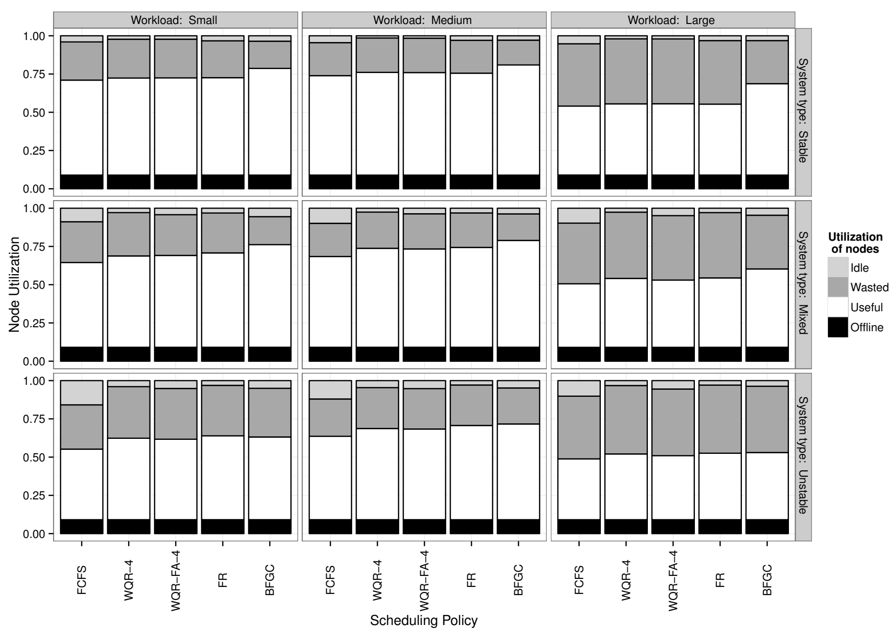


Figure 2: Utilization of nodes for different scenarios (combinations of node stability and task size). Average for all nodes.



(a) Only stable nodes



(b) Only unstable nodes

Figure 3: Utilization of nodes for different scenarios (combinations of node stability and task size) for stable and unstable nodes.

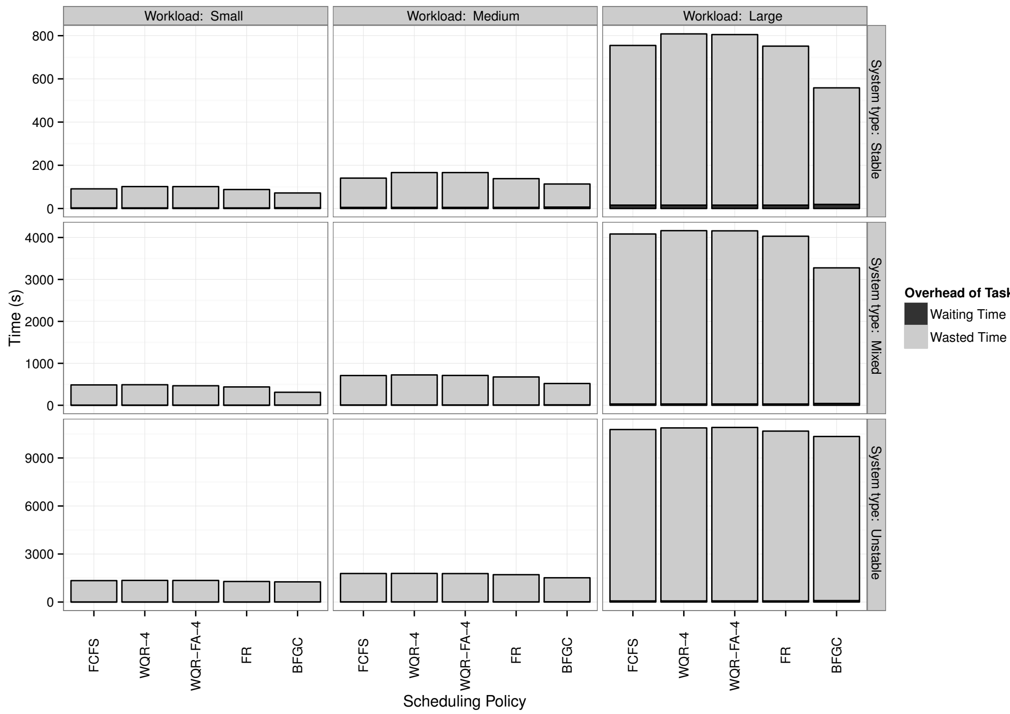


Figure 4: Overheads of tasks for different scenarios (combinations of node stability and task size). Note the different y-axis scale for each row

References

[1] Kondo, D., Javadi, B., Iosup, A., Epema, D.: The Failure Trace Archive: Enabling Comparative Analysis of Failures in Diverse Distributed Systems pp. 398–407 (2010)