

# Dispatch of Labeled/Unlabeled Tasks for Foremen/Multiple-tasks workers

February 6, 2014

## Abstract

Keep unlabeled tasks unlabeled; unlabeled tasks should occupy entire, arbitrary workers. This will allow easier resource accounting, and reduce unforeseen interactions, making work queue easier to analyze and debug.

## 1 Current Implementation

Currently, there are four labeling cases:

---

**All resources are labeled.** The task consumes resources from the worker as labeled.

**At least one resource is unlabeled, cores resource is labeled.** The task consumes resources from the worker as labeled. For unlabeled resources, they are set to zero (i.e., the task uses whatever it can get).

**At least one resource is labeled, cores resource is unlabeled.** The same as the previous case, but cores resource is set to 1.

**All resources are unlabeled.** The task consumes resources equal to an average worker.

---

### General Observations

- The current implementation for unlabeled cores equates number of cores with number of tasks. This is a leftover from the one-task-per-single-core-worker implementation.

### Observations on tasks with all resources unlabeled

- It is tempting to say that a completely unlabeled task should consume the whole worker. However, this would make a single task occupy a whole foreman.
- The 'average worker' is a moving target, and it is difficult to analyze how the system will operate. The problem is that we are labeling tasks according to the resources we have seen so far (e.g., two identical tasks may receive different labels).

## 2 Proposed Behaviour

**All resources are labeled.** Remains the same. The task consumes resources from the worker as labeled.

**At least one resource is unlabeled, cores resource is labeled.** The task consumes resources from the worker as labeled. For unlabeled resources, they are set to -1 (i.e., the task uses whatever it can get).

**At least one resource is labeled, cores resource is unlabeled.** Cores is set to -1, and the task consumes an entire worker in which the labeled resources fit.

**All resources are unlabeled.** The task consumes an entire, arbitrary worker. If dispatching to a foreman, the task consumes an entire worker in that foreman hierarchy.

---

### Observations

- Unlabeled tasks remain unlabeled, since they occupy a whole worker.
- Number of unlabeled tasks would equate to number of workers (rather than cores).
- The idea of 'average worker' is only used when matching (perhaps partially) labeled tasks to foremen.

## 3 Possible Improvements

- Rather than 'average worker', 'median worker' might be a better statistic for a foreman. In the case of outliers, median is stable, and it can also be computed in expected linear time.
- An unlabeled task may occupy the unused resources of a worker already running tasks. The accounting of resources is a complication that we might not want, though.