Reducing accumulated relative tardiness (ART) for soft realtime load balancing

Abstract

Many systems today have incorporated in them the idea of task priorities and have a growing requirement of soft real-time guarantees while making scheduling (load balancing) decisions. In this project, I am looking at the idea of using reinforcement learning to tune weights of the weighted round robin (WRR) load balancer to distribute resource between tasks to minimize the overall tardiness in a system.

Introduction

I am looking at load balancing, possibly in edge clusters, as a soft real time system. This system might have deadlines for different processess or for processes generated by certain sources. But may not have any deadline requirements for many others.

- Observe and collect statistics from the load balancer to create different clusters to broadly define different "states" of the system.
  - This allows us to define a finite number of states.
  - The number of clusters have to be pre-defined. The dimensionality of the data used is a good guide in that direction.
- Once states are determined, use reinforcement learning to find an optimal scheduling solution that looks to reduce tardiness given optimization goals.
  - Optimization goals will be provided in terms of service times.
  - $$\boldsymbol{tardiness} := max[0, (t - d)]$$
    1. t => total time spent in the system by a job
    2. d => deadline for the process.

Project Goal

$$Goal := min \sum_{i=1}^{N(T)} max[0, \frac{(t_{ij} - d_{ij})}{d_{ij}}]$$

So,

$$G_t = max \sum_{i=1}^{N(T)} max[0, -\frac{(t_{ij} - d_{ij})}{d_{ij}}]$$

where,

$$d_{ij} = \mu_{t_{ij}} + c_j \sigma_{t_{ij}}$$

$$c_j = 1 + \frac{1}{p_j}$$

T is the ***time interval at which the reward is measured*** and $N(T)$ is the ***number of jobs that go through that system in that time***.

where:

$$p_{ij} := \text{priority of task type j}$$

$$c_j \in (1, 2)$$

$\mu_{t_{ij}}$is the mean of all tasks of type j completed when the scheduler was observing the system

$\sigma_{t_{ij}}$is the standard deviation of all tasks of type j completed when the scheduler was observing the system

$\frac{(t_{ij} - d_{ij})}{d_{ij}}$ is the ***relative tardiness*** of the job.

We are interested in minimizing the ***accumulated relative tardiness (ART)*** - $\sum_{i=1}^{N(T)} max[0, \frac{(t_{ij} - d_{ij})}{d_{ij}}]$ of the overall system.

States and storing them

The system will be under an initial period of observation while it's under scripted load. These observations are to be used to create a broad definition of "states" which will consist of a bunch of similar actual states of the system. We will add the actual parameters later.

There can be a predefined number of distinct states K, possibly depending on the dimensionality of the data being considered. Each state can be stored as a mean vector $\bar{\mu}$ and a covariance matrix Σ. The states can be recalibrated if the number of observed data points that lie outside all states; $\bar{s} \notin \{\bar{\mu}_k \pm 3\Sigma_k\} \forall k \in [1, K]$; cross a predefined threshold, $N(k)$. Recalibration will include another observation stage.

Load Balancing

The Weighted Round Robin (WRR) forms the basis of the load balancing algorithm. Different types of tasks have different priorities. For each task type, the backend servers can have different weights. The idea is to readjust the weights of the servers, for each task type, so that the overall resources are distributed between the jobs in the system so as to maximize G_t.

Training the RL model

TODO