The resource demand prediction helps in the creation of preconfigured VMs to minimize the response time to the service request. This scheduling model focuses on reducing resource waste; in turn, it will improve the resource utilization. The resource demand prediction and scheduling model will achieve the tradeoff of service response time and resource utilization.

We represent both the user service requests and the configuration of Pools as a pair of (x,y) values, where x represents the CPU and y represents the memory. In the first instance, we took the constant values to represent the service request demand and the pool configuration. These values are ranging from 0 to 100.

A user service request R (xr, yr) can be allotted to a pool P(xp, yp) if:

xp-xr>=0 and yp-yr>=0 ----------- equation (1)

There may be an associated resource waste with a feasible allocation.

We define the resource waste as the difference of (xp-xr)+(yp-yr).

Following is the list of variables used in the problem formulation:

numRequest represents the number of user service requests and numPools represents the number of pools. RPis a matrix of numRequests \* numpools dimensions; these are binary variables (0/1) where each row represents a service request and each column represents a pool. Any element RPij with 1 value denotes that pool (Pj) can process this request (Ri) as it satisfies the equation 1, otherwise this value will be 0.

Wrp is a matrix to represent the aggregated resource waste of the jobs allocation to pools. Considering user service request R (xr, yr) and a pool P (xp, yp), we define the resource waste Wrp of P serving request R as:

Wrp = (xp-xr) + (yp-yr).

We tested the formulation with two different objectives as:

1. The objective is to minimize resource waste defined as:
2. Another objective is to maximize the value, considering the profit of request allocation, cost of pool maintenance and resource waste. Consider the following variables:

A is a column vector representing the assignment of various requests to pools. It is a binary variable with 0/1 values to represent allocation, 1 means successful allocation and the value 0 denotes that this request was rejected for the processing as there may be no space to occupy it.

P is another column vector representing the earnings from serving a job thus, we call it as profit from each user service requests. Both A and P are having numRequests rows.

C is a row vector with numPools columns to represent the maintenance cost of each of the pools.

PC is another row vector with numPools columns, representing a pool capacity in terms of the number of requests.

We define the value as the difference of the total profit and the total of cost and resource waste (P-C-W). The objective function to maximize the value is defined as:

Constraints: Our model achieves the above two different objectives, under the following two constraints.

1. Multiple pools may be eligible for a request allocation by satisfying equation 1, but there is a constraint that but a request will be served by only one pool. We specify this constraint as:

=1

1. Another constraint is that the number of allotted requests to a pool should be less than or equal to the pool capacity. We represent this constraint as::