Load balancing:

Input:

Active servers: {1, 2, …, k}

Load balancing: For all k, {(i1, p\_i1), (i2, p\_i2), …}

Demand: {k1, d1}

LE = {le\_i} forall 0 <= i < k. initialize to zero.

Remap\_buckets = {}

LB\_new = {}

Consider k in increasing order of demand.

Are all s\_k in active servers set

If false:

Remap\_buckets.add(s\_k)

continue

// is this demand satisfied:

demand\_satisfied = true

For x in s\_k:

demand = d\_k \* p\_k

if (demand + LE [k] > d\_k):

demand\_satisied = false

if demand\_satisfied:

Remap\_buckets.add(s\_k)

Continue

Else:

LB\_new.add(LB\_old)

For k in remap\_buckets: // decreasing demand

Get\_new\_rule(demand, LE)

Get\_new\_rule(LE, demand):

Set = {}

Remaning\_demand = demand

For k in reverse(LE.keys()):

Served\_demand = min(1- LE[k], remaining\_demand)

Servers = server \cup {k}

Demands = served\_demand \cup {k}

Remaining\_demand -= served\_demand

Return set