

Mathematical Model (Single Time Interval)

Sets

- $M = \{1 \text{ (t4g.nano)}, 2 \text{ (t4g.medium)}, 3 \text{ (t4g.xlarge)}, 4 \text{ (r8g.large)}, 5 \text{ (c8g.xlarge)}, 6 \text{ (r8g.2xlarge)}, 7 \text{ (c8g.4xlarge)}, 8 \text{ (c8g.8xlarge)}, 9 \text{ (m8g.8xlarge)}, 10 \text{ (r8g.8xlarge)}, 11 \text{ (m8g.12xlarge)}, 12 \text{ (c8g.16xlarge)}\}$
= Set of Machine Types
 - S = Set of Tasks
 - $J = \{1, 2, \dots, |S|\}$ = Index set of potential instances per machine type

Decision Variables

- x_{ijk} : 1 if task $k \in S$ is assigned to instance $j \in J$ of machine type $i \in M$, 0 otherwise
- y_{ij} : 1 if instance $j \in J$ of machine type $i \in M$ is used, 0 otherwise

Parameters

- c_i : cost per hour in *ofmachinetype* \in M\$
- p_i : CPU limit in vCPUs of machine type $i \in M$
- q_i : Memory limit in GiB of machine type $i \in M$
- m_k : CPU requirement in vCPUs of task $k \in S$
- n_k : Memory requirement in GiB of task $k \in S$
- W : size of time window in hrs for which tasks are executing
- $\alpha = 0.1$: instance startup time in hrs (6 min)
- $s_i = \alpha c_i$: startup cost in Dollars of an instance of machine type $i \in M$

Objective Function

$$\min \sum_{i \in M} \sum_{j \in J} (W c_i y_{ij} + s_i y_{ij})$$

Constraints

- $\sum_{k \in S} m_k x_{ijk} \leq p_i, \quad \forall i \in M, j \in J$ (CPU constraint per instance per machine type)
- $\sum_{k \in S} n_k x_{ijk} \leq q_i, \quad \forall i \in M, j \in J$ (Memory constraint per instance per machine type)
- $\sum_{i \in M} \sum_{j \in J} x_{ijk} = 1, \quad \forall k \in S$ (each task can only be assigned to 1 specific instance)
- $y_{ij} \leq \sum_{k \in S} x_{ijk} \leq |S| y_{ij}, \quad \forall i \in M, j \in J$ (link x and y constraint)
- $x_{ijk} \in \{0, 1\}, \quad \forall i \in M, j \in J, k \in S$
- $y_{ij} \in \{0, 1\}, \quad \forall i \in M, j \in J$

Solution (Single Time Interval)

```
In [2]: function check_optimality(m)
        stat = termination_status(m)
        if stat != MOI.OPTIMAL
            println("Solver did not find an optimal solution: $stat")
        end
    end;

In [1]: using CSV, DataFrames

# Read EC2 types
ec2_df = CSV.read("ec2_subset.csv", DataFrame)

M = collect(1:nrow(ec2_df)) # machine types
println("Number of machine types: ", length(M))
machine_name = Dict{i => ec2_df.Type[i] for i in M)

c = Dict{i => ec2_df.Cost[i] for i in M) # cost per hour of machine type
p = Dict{i => ec2_df.vCPUs[i] for i in M) # CPU limit of machine type
q = Dict{i => ec2_df.Memory[i] for i in M) # Memory limit of machine type
W = 3 # time interval window size
α = 0.1 # instance startup time
s = Dict{i => α * c[i] for i in M) # startup cost in $ of machine type
```

```
# Read tasks
tasks_df = CSV.read("tasks_t7.csv", DataFrame)
S = collect(1:nrow(tasks_df)) # task indices
J = collect(1:nrow(tasks_df)) # job indices
println("Number of tasks: ", length(S))
m = Dict{k => tasks_df.cpu_cores[k] for k in S} # CPU cores required by task k
n = Dict{k => tasks_df.mem_gb[k] for k in S}; # Memory required by task k
```

Number of machine types: 12

Number of tasks: 24

```
In [6]: using JuMP, HiGHS
model = Model(HiGHS.Optimizer)

@variable(model, x[i in M, j in J, k in S], Bin) # x[i,j,k] = 1 if task k is assigned to instance j of type i
@variable(model, y[i in M, j in J], Bin) # y[i,j] = 1 if instance j of type i is used

@objective(model, Min, sum(w*c[i]*y[i,j] + s[i]*y[i,j] for i in M, j in J)) # minimize cost

@constraint(model, [i in M, j in J], sum(m[k]*x[i,j,k] for k in S) <= p[i]) # CPU Limit
@constraint(model, [i in M, j in J], sum(n[k]*x[i,j,k] for k in S) <= q[i]) # Memory Limit

@constraint(model, [k in S], sum(x[i,j,k] for i in M, j in J) == 1) # each task is assigned to one instance

@constraint(model, [i in M, j in J], y[i,j] <= sum(x[i,j,k] for k in S)) # instance is active if it has tasks assigned
@constraint(model, [i in M, j in J], sum(x[i,j,k] for k in S) <= length(S)*y[i,j]) # instance can only have tasks assigned if

set_silent(model)
optimize!(model)
check_optimality(model)

println("Minimized total cost: ", objective_value(model))
for i in M
    println("Machine type: ", machine_name[i], ", CPU limit: ", p[i], ", Memory limit: ", q[i])
    println("  Total instances: ", sum(value(y[i, j]) for j in J))
    println("  Total assigned tasks: ", sum(value(x[i, j, k]) for j in J, k in S))
    for j in J
        if value(y[i, j]) > 0.5
            for k in S
                if value(x[i, j, k]) > 0.5
                    println("          Task $k (CPU: ", m[k] , ", Memory: ", n[k] ,") is assigned to instance $j")
                end
            end
        end
    end
end
end
end
```

Minimized total cost: 2.4998399999999995
Machine type: t4g.nano, CPU limit: 2, Memory limit: 0.5
Total instances: 0.0
Total assigned tasks: 0.0
Machine type: t4g.medium, CPU limit: 2, Memory limit: 4.0
Total instances: 15.999999999999995
Total assigned tasks: 15.999999999999927
Task 21 (CPU: 2, Memory: 0.5088) is assigned to instance 1
Task 20 (CPU: 2, Memory: 0.5088) is assigned to instance 2
Task 11 (CPU: 1, Memory: 3.0544) is assigned to instance 3
Task 2 (CPU: 1, Memory: 2.5472) is assigned to instance 4
Task 4 (CPU: 1, Memory: 2.5472) is assigned to instance 5
Task 24 (CPU: 2, Memory: 2.8) is assigned to instance 7
Task 3 (CPU: 1, Memory: 2.5472) is assigned to instance 8
Task 22 (CPU: 2, Memory: 1.22272) is assigned to instance 9
Task 8 (CPU: 1, Memory: 2.5472) is assigned to instance 10
Task 1 (CPU: 1, Memory: 2.5472) is assigned to instance 11
Task 5 (CPU: 1, Memory: 2.5472) is assigned to instance 12
Task 12 (CPU: 1, Memory: 3.3088) is assigned to instance 13
Task 10 (CPU: 1, Memory: 2.6752) is assigned to instance 14
Task 23 (CPU: 2, Memory: 2.5472) is assigned to instance 15
Task 14 (CPU: 1, Memory: 3.8208) is assigned to instance 19
Task 16 (CPU: 1, Memory: 3.8208) is assigned to instance 24
Machine type: t4g.xlarge, CPU limit: 4, Memory limit: 16.0
Total instances: 2.0
Total assigned tasks: 7.999999999999989
Task 13 (CPU: 1, Memory: 3.8208) is assigned to instance 12
Task 15 (CPU: 1, Memory: 3.8208) is assigned to instance 12
Task 17 (CPU: 1, Memory: 4.0704) is assigned to instance 12
Task 19 (CPU: 1, Memory: 4.0704) is assigned to instance 12
Task 6 (CPU: 1, Memory: 2.5472) is assigned to instance 13
Task 7 (CPU: 1, Memory: 2.5472) is assigned to instance 13
Task 9 (CPU: 1, Memory: 2.5472) is assigned to instance 13
Task 18 (CPU: 1, Memory: 4.0704) is assigned to instance 13
Machine type: r8g.large, CPU limit: 2, Memory limit: 16.0
Total instances: 0.0
Total assigned tasks: 0.0
Machine type: c8g.xlarge, CPU limit: 4, Memory limit: 8.0
Total instances: 0.0
Total assigned tasks: 0.0
Machine type: r8g.2xlarge, CPU limit: 8, Memory limit: 64.0
Total instances: 0.0
Total assigned tasks: 0.0
Machine type: c8g.4xlarge, CPU limit: 16, Memory limit: 32.0
Total instances: 0.0
Total assigned tasks: 0.0
Machine type: c8g.8xlarge, CPU limit: 32, Memory limit: 64.0
Total instances: 0.0
Total assigned tasks: 0.0
Machine type: m8g.8xlarge, CPU limit: 32, Memory limit: 128.0
Total instances: 0.0
Total assigned tasks: 0.0
Machine type: r8g.8xlarge, CPU limit: 32, Memory limit: 256.0
Total instances: 0.0
Total assigned tasks: 0.0
Machine type: m8g.12xlarge, CPU limit: 48, Memory limit: 192.0
Total instances: 0.0
Total assigned tasks: 0.0
Machine type: c8g.16xlarge, CPU limit: 64, Memory limit: 128.0
Total instances: 0.0
Total assigned tasks: 0.0

In []: