Performance and Economic Models in the Cloud: Case of Serverless Computing and Spot Instances

1 System

There are two forms of computing available in the cloud: Spot Instance and Serverless Computing.

In **Spot Instances**, the client rents a Virtual Machine (VM), which has a fixed set of physical resources such as CPUs and cores, main memory, long term storage and bandwidth. The Virtual Machine (VM) can run multiple containers. Assume that VMs alternate between idle and busy. There is a startup delay for VM to run containers. The client pays per second for the duration that the VM runs. The price of the VM fluctuates based on supply. The VM only executes as long as the price is below the client's specified price threshold.

In **Serverless Computing**, the cloud provider provisions containers and executes the client's functions. The cost is determined by the function's execution duration (every 100ms) and number of executions.

2 Key Questions

- To reduce the cost, which form of computing should be used?
- Does it make business case to use both forms of computing? If so, how much computing should be done by Serverless? How many VMs should be allocated?

3 Performance Model

To model **Serverless Computing**, assume function calls arrive according to a Poisson process with rate λ_s. Each function runs in the server for a random amount of time, exponentially distributed with rate μ_s.

To model **Spot Instances**, we start with a simpler case with only one VM and fixed price for the VM. Assume there is a setup time, exponentially distributed with rate γ. Assume the functions arrive according to a Poisson process with rate λ_m. Each function runs in the VM for a random amount of time, exponentially distributed with rate μ_m.

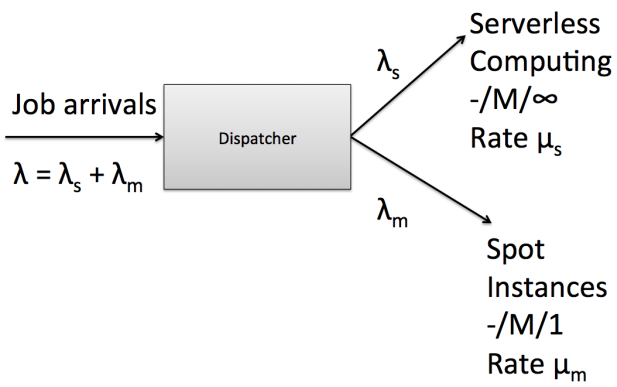

Figure 1: Performance model for VMs and serverless computing in a cloud server.

3.1 Cost Model

Serverless : $C_s(t) = \alpha_s t$ (for each job that takes time t)

Spot Instances: $C_m(t) = \alpha_m t$ (for each VM up for time t including setup time)

3.2 Cost Analysis

In this section, we compute the expected cost per unit time for serverless and spot instances.

3.2.1 Serverless

For serverless, the cost is dependent on the number of function executions $E[Q]$.
Expected cost per unit time = $\alpha_s E[Q] = \frac{\alpha_s \lambda_s}{\mu_s}$

3.2.2 Spot Instances

- *Case 1: VM ON always*

  Expected cost per time = α_m

- *Case 2:**APPROXIMATE** single VM ON only to serve jobs*

  In this case, the expected cost per unit time consists of cost to serve jobs as well as the cost for setting up the first instance.

  Cost to serve jobs = $\frac{\alpha_m \lambda_m}{\mu_m}$

  Computing cost for first instance:
  Time idle = $T - T\rho$ where T is the total time and ρ is busy time of VM
  Number of occurrences when first instance is used (a.k.a. number of times queue is empty) = $\frac{T - T\rho}{\frac{1}{\lambda_m}} = \lambda_m(T - T\rho) = \lambda_m T(1 - \rho)$

Cost for first instance = α_m * number of times queue is empty * setup time cost = $\frac{\alpha_m \lambda_m (1-\rho)}{\gamma}$

Expected cost per unit time = $\frac{\alpha_m \lambda_m}{\mu_m} + \frac{\alpha_m \lambda_m (1-\rho)}{\gamma}$

Substituting ρ with $\frac{\lambda_m}{\mu_m}$,

Expected cost per unit time = $\frac{\alpha_m \lambda_m}{\mu_m} + \frac{\alpha_m \lambda_m (1-\frac{\lambda_m}{\mu_m})}{\gamma}$

$= \frac{\alpha_m \lambda_m}{\mu_m} + \frac{\alpha_m \lambda_m (\mu_m - \lambda_m)}{\mu_m \gamma}$

$= \alpha_m \lambda_m (\frac{1}{\mu_m} + \frac{\mu_m - \lambda_m}{\mu_m \gamma})$

- *Case 3:* ***APPROXIMATE*** *Multiple VMs ON only to serve jobs*

  This case increases the number of VMs that are utilized as opposed to a single VM in case 2. Assume that i is the number of VMs, each of which can be turned ON and OFF. The arrival rate of jobs for each VM is equal, where the rate is $\frac{\lambda_m}{i}$.

  Cost to serve jobs (same as in case 2) = $\frac{\alpha_m \lambda_m}{\mu_m}$

  Cost for first instance for multiple VMs = sum of cost for instance for each VM

  From case 2, cost for first instance for a single VM for arrival rate $\lambda_m = \alpha_m \lambda_m (\frac{1}{\mu_m} + \frac{\mu_m - \lambda_m}{\mu_m \gamma})$

  For an arrival rate $\frac{\lambda_m}{i}$, cost for first instance of a single VM = $\frac{\alpha_m \lambda_m}{i} (\frac{1}{\mu_m} + \frac{\mu_m - \frac{\lambda_m}{i}}{\mu_m \gamma})$

  $= \frac{\alpha_m \lambda_m}{i} (\frac{1}{\mu_m} + \frac{\mu_m i - \lambda_m}{\mu_m \gamma i})$

- *Case 4:* ***EXACT*** *Multiple VMs ON only to serve jobs*
  Assume that i is the number of VMs, each of which can be turned ON and OFF. The arrival rate of jobs for each VM is equal, where the rate is $\frac{\lambda_m}{i}$.

  Cost to serve job for each VM = = $\frac{\alpha_m \lambda_m}{\mu_m i}$

  Cost for starting an instance = $\frac{1}{E[busy]+E[idle]+E[startup]} (\frac{1}{\gamma}) \alpha_m$

  where E[busy] = $\frac{1}{\mu_m - (\frac{\lambda_m}{i})}$, E[idle] = $\frac{1}{\lambda_m}$ and E[startup] = $\frac{1}{\gamma}$

  We simplify the cost for starting an instance as follows:

  - Upper bound cost = $\frac{1}{E[busy]+E[startup]} (\frac{1}{\gamma}) \alpha_m = i(\frac{\alpha_m \lambda_m}{\mu_m i} + \frac{\alpha_m \mu_m i - \alpha_m \lambda_m}{\gamma i + \mu_m i - \lambda_m}) = \frac{\alpha_m \lambda_m}{\mu_m} + \frac{\alpha_m \mu_m i^2 - \alpha_m \lambda_m i}{\gamma i + \mu_m i - \lambda_m}$
  - Lower bound cost = $\frac{1}{2*E[busy]+E[startup]} (\frac{1}{\gamma}) \alpha_m = i(\frac{\alpha_m \lambda_m}{\mu_m i} + \frac{\alpha_m \mu_m i - \alpha_m \lambda_m}{2\gamma i + \mu_m i - \lambda_m}) = \frac{\alpha_m \lambda_m}{\mu_m} + \frac{\alpha_m \mu_m i^2 - \alpha_m \lambda_m i}{2\gamma i + \mu_m i - \lambda_m}$

3.2.3 Total Cost

Question: What are optimal values for λ_s^* and λ_m^* to give minimum cost?
Parameters: λ, α_s, α_m, μ_s, μ_m, γ, i
Decision variables: λ_m^*, λ_s^*

- **Single VM, case 2 in previous subsection**

  The total cost is the sum of the cost of serverless and spot instances.

  Total cost = $\frac{\alpha_s \lambda_s}{\mu_s} + \alpha_m \lambda_m (\frac{1}{\mu_m} + \frac{\mu_m - \lambda_m}{\mu_m \gamma})$
  Substituting λ_s by $\lambda - \lambda_m$ in total cost equation, we get

  Total cost (TC) = $\frac{\alpha_s}{\mu_s}(\lambda - \lambda_m) + \alpha_m \lambda_m (\frac{1}{\mu_m} + \frac{\mu_m - \lambda_m}{\mu_m \gamma})$

  Differentiation w.r.t. λ_m,

  $\frac{dTC}{d\lambda_m} = \frac{-\alpha_s}{\mu_s} + \frac{\alpha_m}{\mu_m} + \frac{\alpha_m}{\mu_m \gamma}(\mu_m - 2\lambda_m)$

  To compute λ_m^*, we set $\frac{dTC}{d\lambda_m} = 0$

  $\frac{-\alpha_s}{\mu_s} + \frac{\alpha_m}{\mu_m} + \frac{\alpha_m}{\mu_m \gamma}(\mu_m - 2\lambda_m) = 0$

  $\lambda_m^* = \frac{-\gamma \mu_m \alpha_s}{2\mu_s \alpha_m} + \frac{\gamma + \mu_m}{2}$

  Constraint: utilization should be less than 1 $\rightarrow \lambda_m < \mu_m$
  $\lambda_s^* = \lambda - \lambda_m^* = \lambda + \frac{\gamma \mu_m \alpha_s}{2\mu_s \alpha_m} - \frac{\gamma + \mu_m}{2}$

- **Multiple VMs (Approximate, case 3 in previous subsection**

  The total cost is the sum of the cost of serverless and spot instances.

  Total cost = $\frac{\alpha_s \lambda_s}{\mu_s} + \frac{\alpha_m \lambda_m}{i} (\frac{1}{\mu_m} + \frac{\mu_m i - \lambda_m}{\mu_m \gamma i})$
  Substituting λ_s by $\lambda - \lambda_m$ in total cost equation, we get

  Total cost (TC) = $\frac{\alpha_s}{\mu_s}(\lambda - \lambda_m) + \frac{\alpha_m \lambda_m}{i} (\frac{1}{\mu_m} + \frac{\mu_m i - \lambda_m}{\mu_m \gamma i})$

  Differentiation w.r.t. λ_m,

  $\frac{dTC}{d\lambda_m} = \frac{-\alpha_s}{\mu_s} + \frac{\alpha_m}{\mu_m i} + \frac{\alpha_m}{\mu_m \gamma i^2}(\mu_m i - 2\lambda_m)$

  To compute λ_m^*, we set $\frac{dTC}{d\lambda_m} = 0$

  $\frac{-\alpha_s}{\mu_s} + \frac{\alpha_m}{\mu_m i} + \frac{\alpha_m}{\mu_m \gamma i^2}(\mu_m i - 2\lambda_m) = 0$

  $\lambda_m^* = \frac{-\gamma \mu_m \alpha_s i^2}{2\mu_s \alpha_m} + \frac{(\gamma + \mu_m) i}{2}$

  Constraint: utilization should be less than 1 $\rightarrow \lambda_m < \mu_m$
  We add an additional constraint: $\lambda_m < \beta\lambda$, where $\beta = 0.8$
  $\lambda_s^* = \lambda - \lambda_m^* = \lambda + \frac{\gamma \mu_m \alpha_s i^2}{2\mu_s \alpha_m} - \frac{(\gamma + \mu_m) i}{2}$

- **Multiple VMs (EXACT, case 4 in previous subsection**

  - Total cost for Upper bound = $\frac{\alpha_s \lambda_s}{\mu_s} + \frac{\alpha_m \lambda_m}{\mu_m} + \frac{\alpha_m \mu_m i^2 - \alpha_m \lambda_m i}{\gamma i + \mu_m i - \lambda_m}$
  - Total cost for Lower bound = $\frac{\alpha_s \lambda_s}{\mu_s} + \frac{\alpha_m \lambda_m}{\mu_m} + \frac{\alpha_m \mu_m i^2 - \alpha_m \lambda_m i}{2\gamma i + \mu_m i - \lambda_m}$

4 Consulting model

A VM can be thought of as an employee - costs less but you have to pay them all the time. Serverless is like a consultant, expensive but only pay for the hours worked.

For systems with dynamic workloads, we can think of threshold policies with the response time as the constraint (which works out a constraint on utilization). Suppose your system is running at capacity with N VMs. Now if a little extra load comes in, it is cheaper to send it to serverless, until the excess load becomes enough so that N+1 VMs are more cost effective. The same thing applies in the reverse direction - if the load reduces, it becomes cost effective to shut down a VM and send the excess load to serverless, until the load reduces enough to rebalance amongst the N-1 servers and maintain the utilization.

4.1 Forward case

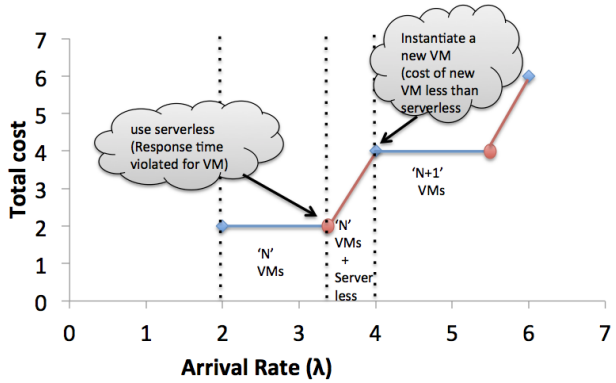

Figure 2: Serverless is like Consulting

Assume that the entire workload (λ) is executed on N machines. In this case, the workload is increasing.

4.1.1 When to send jobs to serverless

Extra workload (ϵ) comes in. We need to calculate the value of the extra workload (ϵ) for which we need to send jobs to serverless. We send jobs above the value of ϵ to serverless. We send the jobs to serverless under the condition that the active VMs are unable to meet the response times for the workload. The response times will be violated when the response time of the current workload exceeds the response time requirement ($\frac{1}{r_{user}}$), which is a constant, specified by the user.

Total workload $= \lambda + \epsilon$
Average response time for M/M/1 queue $= \frac{1}{\mu_m - (\lambda+\epsilon)} = \frac{1}{\mu_m - \lambda - \epsilon}$
Average response time violation condition : Average response time $> \frac{1}{r_{user}}$
To calculate the value of extra workload (ϵ): $\frac{1}{\mu_m - \lambda - \epsilon} = \frac{1}{r_{user}}$
$\epsilon = \mu_m - \lambda - r_{user}$

If the extra workload >= ϵ, we start sending jobs to serverless until we decide that it is cheaper to instantiate a new VM and execute the entire workload on the VMs.

4.1.2 When to instantiate a new VM

Extra workload (ϵ') comes in, where ($\epsilon' > \epsilon$). We need to calculate the value of the extra workload (ϵ') for which it is cheaper to instantiate a new VM, such that all jobs are executed on the VM.

For a time interval T,

Cost for VM (always ON) = $\alpha_m T$

Cost for serverless in T = $\alpha_s \times$number of jobs arrived in T $\times$ response time per job = $\alpha_s(\epsilon' T)(\frac{1}{\mu_s})$

To calculate the value of extra workload (ϵ'), cost for VM = cost for serverless

$\alpha_m T = \alpha_s(\epsilon' T)(\frac{1}{\mu_s})$

$\epsilon' = \frac{\alpha_m}{\alpha_s}\mu_s$

If the extra workload >= ϵ', we instantiate a new VM.

4.2 Backward case

Assume that the entire workload (λ) is executed on N machines. In this case, the workload is decreasing and we might be better off by shutting down a VM. We need to calculate two thresholds: 1) when to shut down a VM such that the workload is executed by $N-1$ VMs and serverless 2) when to shut down a VM such that the entire workload is executed by $N-1$ VMs.

4.2.1 When to shut down a VM such that the entire workload is executed by $N-1$ VMs.

Assume that the workload is reduced by an amount σ in a time interval T. In this case, the N VMs are underutilized and we can use $N-1$ to execute the entire workload such that the response time (r_{user}) requirement is not violated. Essentially, this case boils down to calculating the maximum workload that will lead to maximum utilization of $N-1$ VMs.

Total workload = $\lambda - \sigma$ Average response time = $\frac{1}{\mu_m - (\lambda - \sigma)} = \frac{1}{\mu_m - \lambda + \sigma}$

The maximum workload will be obtained when the average response time is equal to the response time requirement.

$\frac{1}{\mu_m - \lambda + \sigma} = \frac{1}{r_{user}}$

$\sigma = r_{user} - \mu_m + \lambda$

4.2.2 When to shut down a VM, where the workload is executed by $N-1$ VMs and serverless

Assume that the workload is reduced by an amount σ' (where $\sigma' < \sigma$) in a time interval T. We want to shut down a VM when the cost of running N VMs is greater than the cost of $N-1$ VMs plus serverless. Note that at this threshold value, the N VMs will be underutilized and we can pack jobs in N VMs with a higher utilization.

Cost of N VMs $= N\alpha_m T$

Cost of $N-1$ VMs $= (N-1)\alpha_m T$

Is this correct?Cost of serverless for time $T = \alpha_s \times$ number of jobs in T $\times$ response time per job
$= \alpha_s \times ((\lambda - \sigma) - (\lambda - \sigma'))T \times (\frac{1}{\mu_s})$

To calculate the threshold value, cost of N VMs = cost of $N-1$ VMs + cost of serverless

$N\alpha_m T = (N-1)\alpha_m T + \alpha_s \times ((\lambda - \sigma) - (\lambda - \sigma'))T \times (\frac{1}{\mu_s})$

$\sigma' = (\frac{\alpha_m}{\alpha_s})\mu_s + \sigma$