Swayam: Distributed Autoscaling to Meet Service Level Agreements of Machine Learning Inference Services with Resource Efficiency

Authors anonymized for review

Appendix

Given a load balancing policy that dispatches requests to random backends,

Swayam must proactively scale-out backend servers based on predicted load. Estimating a sufficient number of backends required to handle a given load is thus an integral part of Swayam's model. Since the objective is to meet response time SLAs, e.g., at least 99% of requests should complete under 100 ms, the resource estimator must account for waiting times due to the underlying load balancing (LB) policy, request processing times, and the desired service level, i.e., "99%" in the above example.

We propose an *approximate analysis* to derive percentile response times, assuming that a request is immediately dispatched to a random backend upon arrival.

policy (defined below). which takes as input the desired service level and the distribution of the request processing times, and given that requests are immediately dispatched to a random backend upon arrival.

Assumption 1. *The average time to send a message from any frontend to any backend is d_1 and the average time to send a message from any backend to any frontend is d_2. Requests arrive at an average rate of λ and that the average time to service a request is $1/\mu$.*

Definition 1. ***RAND LB Policy:*** *Upon receipt of a request from the broker, the frontend immediately forwards it to a random backend. If the backend is busy, it sends back a busy response to the frontend. Upon receipt of a busy response, the frontend forwards the request to another random backend after Δ time units.*

Theorem 1 (Waiting Time Distribution)**.** *If there are n backends, and requests are assigned to backends as per the **RAND LB Policy**, then the expected p^{th} percentile waiting time of the requests can be approximated as:*

$$\omega_p = d_1 + \left(\frac{\ln\left(1 - \frac{p}{100}\right)}{\ln\left(\frac{\lambda}{n \cdot \mu}\right)} - 1 \right) \cdot (d_1 + d_2 + \Delta). \quad (1)$$

Proof. Assume that $d_1, d_2, \Delta \ll 1/\mu$. As a result, the duration for which the system remains non-work-conserving is negligible. Thus, the probability that any backend is busy at any given point of time is given by the overall system utilization:

$$P_{busy} = \frac{\lambda}{n \cdot \mu}. \quad (2)$$

Similarly, let the probability that any backend is idle at any given point of time be denoted as:

$$P_{idle} = 1 - P_{busy} = 1 - \frac{\lambda}{n \cdot \mu}. \quad (3)$$

Let P_r denote the probability that a request finds an idle backend after r retries, i.e., P_r denotes the probability that a request is assigned to busy backends during the first r attempts, and only in the $r+1^{st}$ retry, the request is assigned to an idle backend. Thus,

$$P_r = (P_{busy})^r \cdot P_{idle}. \quad (4)$$

Let W_r denote the waiting time of the request that finds an idle backend after r retries. W_r consists of r round-trip latencies for the first r attempts, r delays of time Δ each enforced by the frontend, and a single frontend-to-backend communication latency for the last successful attempt. Thus,

$$\begin{aligned} W_r &= r \cdot (d_1 + d_2) + r \cdot \Delta + d_1 \\ &= r \cdot (d_1 + d_2 + \Delta) + d_1. \end{aligned} \quad (5)$$

Let ω_p denote the p^{th} percentile waiting time and suppose that it consists of delays due to r_{max} retries. Thus,

$\omega_p = W_{r_{max}}$, i.e.,

$$\omega_p = r_{max} \cdot (d_1 + d_2 + \Delta) + d_1$$
$$\equiv \quad r_{max} = \frac{\omega_p - d_1}{d_1 + d_2 + \Delta}. \tag{6}$$

If any request requires less than or equal to r_{max} retries to find an idle backend, then its waiting time is also less than or equal to ω_p. Therefore:

$$\sum_{k=0}^{r_{max}} P_k = \frac{p}{100}.$$
$$\equiv \sum_{k=0}^{r_{max}} ((P_{busy})^k \cdot P_{idle}) = \frac{p}{100}.$$
$$\equiv P_{idle} \cdot \sum_{k=0}^{r_{max}} (P_{busy})^k = \frac{p}{100}.$$

{by using sum of geometric progression}

$$\equiv P_{idle} \cdot \frac{1 - (P_{busy})^{r_{max}+1}}{1 - P_{busy}} = \frac{p}{100}.$$

{from Eq. 3 }

$$\equiv 1 - (P_{busy})^{r_{max}+1} = \frac{p}{100}.$$

{from Eq. 2 }

$$\equiv 1 - \left(\frac{\lambda}{n \cdot \mu}\right)^{r_{max}+1} = \frac{p}{100}.$$
$$\equiv \left(\frac{\lambda}{n \cdot \mu}\right)^{r_{max}+1} = 1 - \frac{p}{100}.$$
$$\equiv r_{max} + 1 = \frac{\ln\left(1 - \frac{p}{100}\right)}{\ln\left(\frac{\lambda}{n \cdot \mu}\right)}.$$

{from Eq. 6 }

$$\equiv \frac{\omega_p - d_1}{d_1 + d_2 + \Delta} + 1 = \frac{\ln\left(1 - \frac{p}{100}\right)}{\ln\left(\frac{\lambda}{n \cdot \mu}\right)}.$$
$$\equiv \omega_p = d_1 + \left(\frac{\ln\left(1 - \frac{p}{100}\right)}{\ln\left(\frac{\lambda}{n \cdot \mu}\right)} - 1\right) \cdot (d_1 + d_2 + \Delta)$$

Hence, Eq. 1 is proved. □

Theorem 2 (Response Time Distribution)**.** *If there are n backends, if the requests are assigned to backends as per the* ***RAND LB Policy****, and the CDF of the request execution times is denoted by $CDF_{exe}(x)$, then the probability that the response time of a request is less than or equal to RT is given by*

$$\sum_{r=0}^{r_{max}} \left(\frac{\lambda}{n \cdot \mu}\right)^r \cdot \left(1 - \frac{\lambda}{n \cdot \mu}\right) \cdot CDF_{exe}(RT - WT_r), \tag{7}$$

where $r_{max} = \lfloor (RT - d_1)/(d_1 + d_2 + \Delta) \rfloor$ and $WT_r = r \cdot (d_1 + d_2 + \Delta) + d_1$.

Proof. Let Y and Z denote the random variables corresponding to request execution times and waiting times, respectively. Let $R = Y + Z$ denote the random variable corresponding to the request response time. Thus, by the general formula for the distribution of the sum of two independent discrete variables:

$$P(R \leq RT) = \sum_{z=-\infty}^{\infty} P(Z = z) \cdot P(Y \leq RT - z)$$

{since Z can only take discrete values based on the number of retries, i.e., for r retries, $Z = WT_r = r \cdot (d_1 + d_2 + \Delta) + d_1$, and since r varies from 0 to ∞}

$$P(R \leq RT) = \sum_{r=0}^{\infty} P(Z = WT_r) \cdot P(Y \leq RT - WT_r)$$

{since R, Y, Z are non-negative, $R \leq RT$ implies that $Z = WT_r \leq RT$, which in turn implies that $r \leq r_{max} = \lfloor (RT - d_1)/(d_1 + d_2 + \Delta) \rfloor$}

$$P(R \leq RT) = \sum_{r=0}^{r_{max}} P(Z = WT_r) \cdot P(Y \leq RT - WT_r)$$

{since $P(Z = WT_r)$ is equivalent to $P_r = (P_{busy})^r \cdot P_{idle}$ in Eq. 4, and since $P_{busy} = (\lambda)/(n \cdot \mu)$ and $P_{idle} = 1 - (\lambda)(n \cdot \mu)$ for the RAND LB Policy}

$$P(R \leq RT) = \sum_{r=0}^{r_{max}} \left(\frac{\lambda}{n \cdot \mu}\right)^r \cdot \left(1 - \frac{\lambda}{n \cdot \mu}\right) \cdot P(Y \leq RT - WT_r)$$

{since $P(Y \leq RT - WT_r)$ is equivalent to the CDF of the execution time distribution}

$$P(R \leq RT) = \sum_{r=0}^{r_{max}} \left(\frac{\lambda}{n \cdot \mu}\right)^r \cdot \left(1 - \frac{\lambda}{n \cdot \mu}\right) \cdot CDF_{exe}(RT - WT_r).$$

Hence, Eq. 7 is proved. □