How to reduce the number of email server lookup requests while keeping error low ?

Quentin Guilloteau
Grenoble INP - Inria Grenoble

Abstract

More and more people are using email clients everyday. But these clients are using strategies to look for new emails that are generating high traffic. In this paper, we present some strategies to reduce the number of requests while keeping the time between a sent email and its reception as low as possible.

1 Introduction

In the past decade, the number of people working in desk-jobs has greatly increased. With this, the need to communicate using Emails. Emails are very useful to share information with people geographically far from us.

However, in order to know if we receive a new email, we need to ask a central server. These requests are managed by your client (web client or application). The strategy used by these clients is to send a request to the server with a given time interval. For example, every five minute the client will ask the server for new messages.

If we keep our mail client open during the day, this could generate a significant number of requests. Moreover, most of the time, the server will respond that there are no new emails.

Let us take an example. Imagine you start your mail client at 9am and close it at 6pm. Your client sends a request to the server every 5 minutes. The request will require some TCP magic, and the total exchange will be around 2 kB (2415 B in my case). So at the end of the day, your client would have been responsible for 260 kB. This would represent 1304 kB per work week (i.e. 5 days), 5 GB per month and 62 GB yearly.

A longer time interval between two server requests will also decrease the reactivity to an email. Indeed, if someone sends me an email between two server requests, I have to wait the next request to know that I receive an email. There is thus a tradeoff to find between the number of requests sent and the time between a mail is sent to us and the time we receive it.

In this paper, we will present some options to reduce the number of requests sent to the server while keeping the error low. We will start by doing an analysis of the emails I have received to extract information for our next models. We will continue by finding an approximation of the optimal time interval between two server request. Then we will look into a probiblistic approach and finish with a solution involving control theory.

2 Email Distribution

We will take a look at the emails I have received on my school account during my scholarship.

By looking a the data, I have received 2209 emails over a 948 day span.

2.1 Distribution of number of emails received everyday

In average, I receive 2.3301688 mails per day with the distribution in Figure 1.

It makes sense to try to fit an exponential distribution on this data.

As a first estimation, we can take the value at 0 to get the parameter of the distribution ($\lambda = 0.3037975$)

2.2 Email Distribution during the Day

We now want to know when do we receive emails during the day.

Figure 2 depicts the time of reception of the emails during the day.

We can see two spikes, they are around 10am and 3pm.

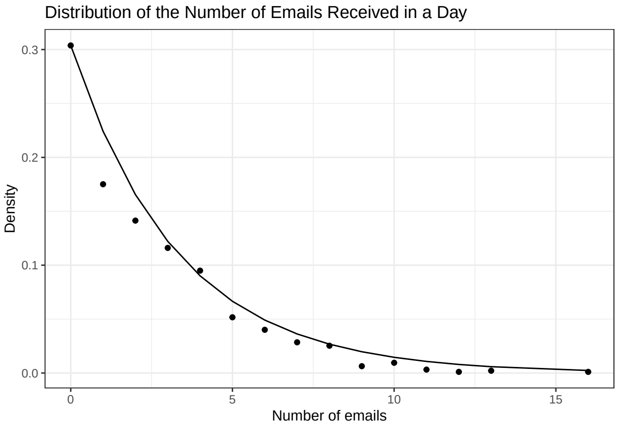

Figure 1: Exponential Distribution for number of mails in a day

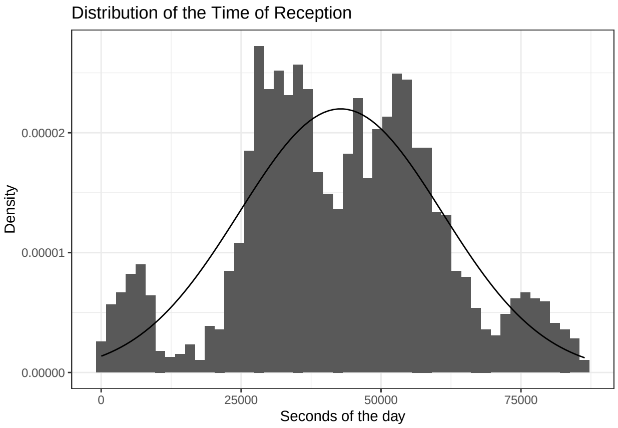

Figure 2: Time of reception of the emails during a day

As an approximation, we can fit a normal distribution over the data. The time of reception of an email will thus follow the distribution $\mathcal{N}(\mu, \sigma^2)$. With $\mu = 42802$ and $\sigma = 18138$.

2.3 Mean Time between two Emails

Now that we know that the distribution of emails received during the day is normal ($\mathcal{N}(\mu, \sigma^2)$), we can compute the mean time between two received emails.

Let $T_1, T_2 \rightsquigarrow \mathcal{N}(\mu, \sigma^2)$. We want to compute $\mathbb{E}[|T_1 - T_2|]$.

So let $Y = |T_1 - T_2|$. The distribution function of Y (whuber 2015) is:

$$f_Y(x) = \frac{1}{\sigma\sqrt{\pi}} e^{-\frac{1}{4}\left(\frac{x}{\sigma}\right)^2} \tag{1}$$

Thus we have the mean of Y:

$$\mathbb{E}[Y] = \int_0^{+\infty} x f_Y(x) dx = \frac{2\sigma}{\sqrt{\pi}} \tag{2}$$

With our distribution, we thus have a mean time between two received emails of 20466.6811136, which represents 5.685 hours.

2.4 Summary

To summarise our observations:

- I receive a number of emails per day following an exponential distribution $Exp(\lambda)$ with $\lambda = 0.3037975$.
- The time of reception of an email during the day can be model by a normal distribution $\mathcal{N}(\mu, \sigma^2)$ with $\mu = 42802$ and $\sigma = 18138$.
- The average time between two received emails is 20466.6811136seconds.

3 Problem Definition

The problem that we are looking at is to reduce the number of requests to the server, while having the time between an email is sent to us and the time we receive it (next request) low.

There are multiple possible ways to model this problem, we decided to model it as follow:

There are N emails sent where $N \rightsquigarrow Exp(\lambda)$.

Let $(T_j)_j$ be the send times of the emails, $\forall\ 1 \leq j \leq N, T_j \rightsquigarrow \mathcal{N}(\mu, \sigma^2)$.

We operate in seconds and during one day. The minimal value is thus 0 and the maximal is 86400. We define $T_0 = 0$ and $T_N = 86400$.

Let $(U_n)_n$ be the times of the requests associated with the chosen requests strategy.

We define J, our cost function, as:

$$J = \sum_{\substack{n \in \mathbb{N} \\ U_n \in [T_0, T_N]}} (U_n - T(U_n))^2 \tag{3}$$

with:

$$T(U_n) = \begin{cases} T_i, & \text{if } \exists i \in [\![1, N]\!], i = \min\{j, U_{n-1} < T_j \leq U_n\} \\ T_i, & \text{otherwise, } i \text{ such that}, T_i \leq U_n \leq T_{i+1} \end{cases} \tag{4}$$

The quantity $(U_n - T(U_n))$ represents the error. A low value means that the time between the last request and the reception of the email is low.

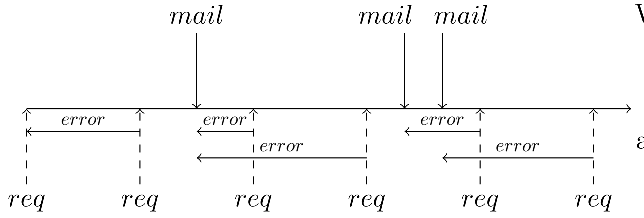

Figure 3: Example for computing the error

An example is depicted in Figure 3.

This cost function has the advantage of penalizing strategies when there are sending too much requests without new email received.

4 Optimal Fixed Time Interval

In this section, we want to find the optimal time interval for our email distributions.

4.1 Definition of the Problem

In the case of the fixed time interval strategy, $U_n = n \times k$ where k is the time interval between two requests, and $U_0 = 0$.

We can now write the cost function J as:

$$J = \sum_{j=1}^{N+1} \left[R_j^2 + \sum_{i=1}^{M_j} (R_j + ik_N)^2 \right] \tag{5}$$

with $M_j = \left\lfloor \frac{T_j}{k} \right\rfloor - \left\lceil \frac{T_{j-1}}{k_N} \right\rceil$.

We define:

- k_N is the time interval for N received emails
- $R_j = \left\lceil \frac{T_{j-1}}{k_N} \right\rceil k_N - T_{j-1}$.
- $J_1 = M_j R_j^2$
- $J_2 = 2kR_j \frac{M_j(M_j+1)}{2}$
- $J_3 = \frac{k_N^2}{6} M_j(M_j+1)(2M_j+1)$
- $\Delta T_j = T_j - T_{j-1}$.

We can expand J:

$$J = \sum_{j=1}^{N+1} \left(R_j^2 + J_1 + J_2 + J_3 \right)$$

We want to minimize J. So we will find a lower bound of J and find the value of k that minimize it.

We remind the reader that:

$$x - 1 < \lfloor x \rfloor \leq x$$

and

$$x \leq \lceil x \rceil < x + 1$$

4.2 Lower Bound of J

Let us find a lower bound for each of these quantities:

- $M_j j > \frac{1}{k_N} \Delta T_j$
- $R_j > 0$
- $J_1 > 0$
- $J_2 > 0$
- $J_3 > \Delta T_j \left(\frac{1}{3k_N} (\Delta T_j)^2 + \frac{1}{2} (\Delta T_j) + \frac{k_N}{6} \right)$

Let us sum everything:

$$\begin{aligned} J &= \sum_{j=1}^{N+1} \left(R_j^2 + J_1 + J_2 + J_3 \right) \\ &> \sum_{j=1}^{N+1} \left(\frac{1}{3k_N} \Delta T_j^3 + \frac{1}{2} \Delta T_j^2 + \frac{k_N}{6} \Delta T_j \right) \\ &= L(k_N) \end{aligned} \tag{6}$$

We want to minimize, so let us differentiate the lower bound with respect to k_N:

$$\begin{aligned}\frac{dL}{dk_N}(k_N) &= \sum_{j=1}^{N+1}\left(\frac{-1}{3k_N^2}\Delta T_j^3 + \frac{1}{6}\Delta T_j\right) \\ &= \frac{1}{6}T_{N+1} - \frac{1}{3k_N^2}\sum_{j=1}^{N+1}\Delta T_j^3\end{aligned} \tag{7}$$

Let us solve for $\frac{dL}{dk_N}(k_N) = 0$:

$$k_N^* = \sqrt{\frac{2\sum_{j=1}^{N+1}\Delta T_j^3}{T_N}} \tag{8}$$

N is the number of emails received this day. T_{N+1} is the end of the day.

4.3 Analysis

There is no easy way to compute the sum of the $(\Delta T_j)^3$. Consequently, we will approximate this value by a simulation.

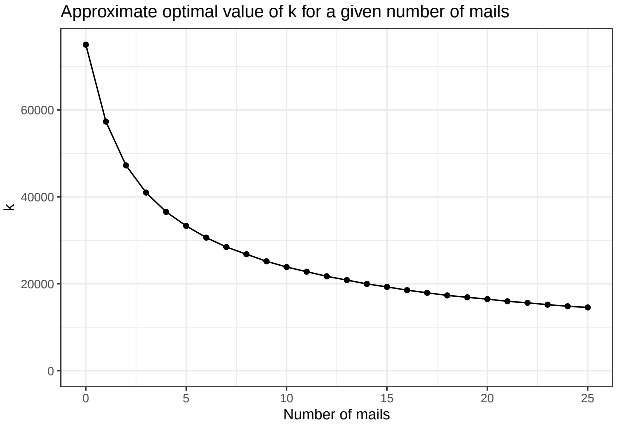

Figure 4: Approximation of the optimal value of k_N

But we do not know how many mail we will receive each day.

But we know the distribution of the number of emails.

Let us compute the weighted mean for the value of k_N^*:

$$k^* = \sum_{n=0}^{+\infty} k_n^* \times P(N = n)$$

where $N \rightsquigarrow Exp(\lambda)$.

From the estimated value of λ seen above, we get that $k^* = 58902$.

This means a request every 981 minutes, for a total of 2 checks everyday.

4.4 Validation

We can simulate typical days for different values of k and compute the mean error during these days. We can then compare the theorical value of k minimizing the error found previously versus the experimental value.

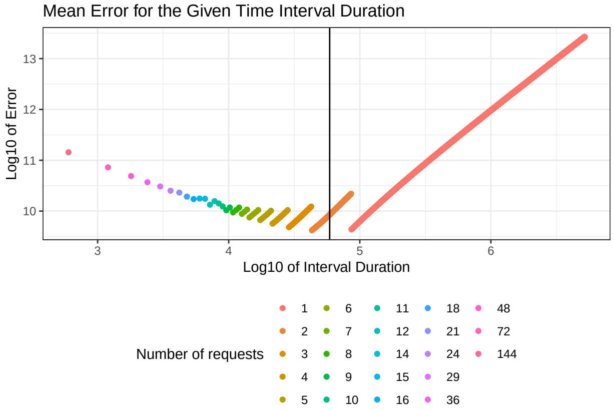

Figure 5: Mean Error for a given Time Interval

We can see on figure 5 that the theoretical optimal value of k^* (vertical line) is close to the experimental optimal.

We can also see that there are clusters of points. Each segment correspond to a certain number of requests. For instance the last segment corresponds to time intervals requireing only one request per day.

In our case, we see that the minimum for the error is reached for the lowest time interval requireing two requests per day. This value of k^* would be 43200 i.e. a request every 12 hours.

4.5 Summary

In this section, we manage to find the optimal time interval for our email distribution.

To summarise the process to finding this optimal value, here is a little guide:

1. Determine the distribution of the number of emails received per day
2. Determine the distribution of emails during the day

3. Compute the approximate value of k_N^* as in Equation 8
4. Compute the weighted sum of the k_N^*
5. With this value of k^*, compute how many requests are required in a day
6. Compute the lowest value of k for this number of requests per day

For the remaining of this paper, we will compare the other approach to the constant time interval method with the optimal value of k^* for our distribution ($k^* = 43200$).

5 Probabilistic Approach

5.1 Principle

We saw previsouly that the distribution of emails during the day could be represented as a normal distribution. In this section, we define a strategy using this distribution.

The idea is to have the time interval adapt in function of the time of the day. The closer we are to the mean (top of the bell) the more the time interval will decrease.

Let M be the maximal time interval and m be the minimal time interval. This means that at all time, the time interval will be between m and M.

If f is the probability density function of the normal distribution $\mathcal{N}(\mu, \sigma^2)$, then let g be the function taking the current time and returning the value of the interval between two requests:

$$\forall x \in [0, 86400], g(x) = M - (M - m)\frac{f(x)}{f(\mu)} \qquad (9)$$

Thus,

$$\forall n \geq 0, U_{n+1} = U_n + g(U_n) \qquad (10)$$

with $U_0 = 0$.

5.2 Simulations

We ran some simulations with this model and compared the results with the optimal time interval value found previously.

The mininal time interval was $m = 900$, the maximal time interval was $M = 20466$. We choose M as the mean time between two received emails as defined in section 2.3.

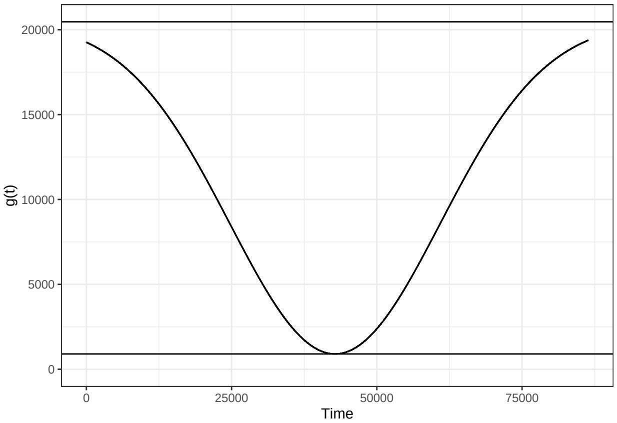

Figure 6: Plot of g

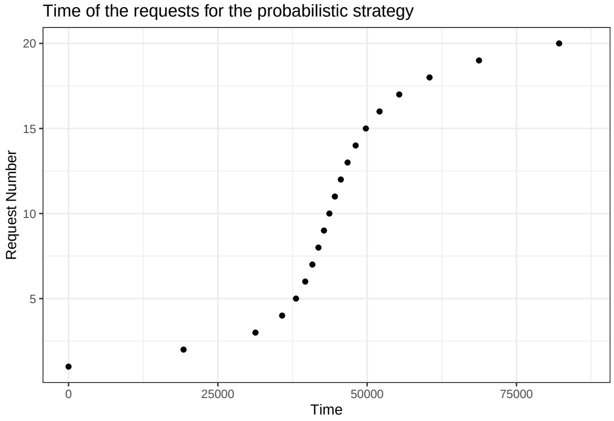

Figure 7: Instants of the requests during the day using the probabilistic approach

We define the error gain for this model as:

$$G_{err} = \frac{Err_k}{Err_{prob}}$$

and the request gain as:

$$G_{req} = \frac{Req_k}{Req_{prob}}$$

We thus have the following 95% Confidence Intervals:

- vs. $\Delta T = k^*$:
  - $G_{req} = 0.0952381$
  - $G_{err} \in [0.3254921, 0.3420143]$
- vs. $\Delta T = 10$mins:
  - $G_{req} = 6.8571429$
  - $G_{err} \in [6.4311307, 6.5092699]$
- vs. $\Delta T = 30$mins:
  - $G_{req} = 2.2857143$
  - $G_{err} \in [2.1882793, 2.2149029]$

5.3 Summary

This approach managed to reduce the number of requests sent and the error for time intervals of 10 and 30 minutes. It did not manage to improve compared to time interval of k^*. These results are strongly linked to the values of m and M.

The befenits of this method is that we send less requests during time of the day where there is less activity, and we send more requests when there is more activity. This allows the user to still be reactive during activity peaks while decreasing the number of requests.

6 PID Controller

6.1 Presentation

The goal is now to design a PID Controller able to regulate the time period between two requests.

The idea is the following: if I have no mail, I increase the time interval, if I just received a mail, then I decrease the time interval.

The expression of a PID Controller is as follow:

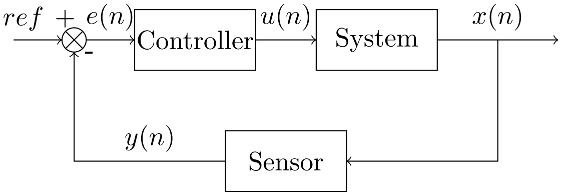

Figure 8: Example of feedback loop

$$u(n+1) = u(n) + k_p e(n) + k_i \int_0^n e(x)dx + k_d \frac{de}{dt}(n) \qquad (11)$$

$e(n)$ is the error at step n and $u(n)$ is the time interval between two requests at step n.

We will define the error as the time since the last email received.

The sign of the error is positive if we did not receive a mail since our last check, negative otherwise.

6.2 Experiments

We designed the experiments to test every controller within a range of values of the k_p, k_i, k_d. We then computed the gains for the error and for the number of requests.

We also had to choose the minimal and maximal time intervals. We decided to take $m = 900$ and $M = 20466$. M is the mean time between two emails as defined in section 2.3.

The parameters that gave us the best overall gains are:

- $k_p = 0.1$
- $k_i = 0.1$
- $k_d = 0$

We then did a full comparison of our best PID controller against intervals of k^*, 10mins and 30mins.

The experiments gave us the following 95% Confidence Intervals:

- vs. $\Delta T = k^*$:
  - $G_{req} \in [0.1922474, 0.1940818]$
  - $G_{err} \in [0.6613566, 0.726482]$
- vs. $\Delta T = 10$mins:

  - $G_{req} \in [13.8418129, 13.9738918]$
  - $G_{err} \in [16.9567991, 17.7957734]$
- vs. $\Delta T = 30$mins:
  - $G_{req} \in [4.6139376, 4.6579639]$
  - $G_{err} \in [5.7592125, 6.0404659]$

6.3 Summary

Using a PID managed to decrease the error and the number of requests compared to intervals of 10 and 30 minutes. It is less efficient than an interval of k^*. However, it managed to have an error inferior to 150% of the error for k^*.

The PID Controller approach also performed better than the probabilistic approach. Indeed, it managed to have higher gains in error and in the number of requests compared to the probabilistic approach.

7 Conclusion

In this paper, we presented multiple strategies to reduce the number of requests sent to the email server while trying to keep the error low. First, we computed the optimal time interval for the classic periodical approach. Then, we presented an approach using the probabilistic distribution of the email reception during the day. This strategy managed to improve the error and the number of requests sent compared to non-optimal standard time intervals. Finally, we introduced a approach using control theory with a PID Controller. This final approach showed far better results than the probabilistic approach, but still was unable to beat the classical periodical approach with the optimal time interval.

We remind the reader that the data from this paper comes from my personal emails and that the results are only valid for my emails. However, the methods should be reproducible.

References

whuber. 2015. "Distribution of Difference Between Two Normal Distributions." https://stats.stackexchange.com/q/186545.