COMPUTER SYSTEMS MODELLING AND SEMANTIC WEB

PRABHJOT PRABHJOT

SOCIAL MEDIA PUBLICATIONS

A newsletter dispatching systems, receives two kinds of streams:

- a quick news information stream (on the average, **λ_news = 2 news/second**);
- a detailed article stream (on the average **λ_articles = 0.1 articles/second**).

  Each stream is processed by a text and image processing systems, where quick news requires an average of D_news = 0.25 sec while articles need an average of D_articles= 3 seconds. Feeds are then sent to three social media, namely Facebook, Twitter and Instagram, each one using a sending queue that works in FCFS. Each feed might be sent to one or more of the considered media. In particular, each social network receives the following percentage of feeds, and requires a different average time for being processed:

Social media	news %	news time	article %	article time
Facebook	50%	8 s.	80%	10 s.
Twitter	80%	6 s.	10%	12 s.
Instagram	75%	8 s.	60%	9 s.

All service time distributions can be considered exponential, and arrivals are Poisson processes. Note that social media can be modeled by infinite server stations, since those services are generally spread over an extremely large number of servers (which is unknown to the users). The administrator would like to study the average time required to publish an article over all its media.

OVERVIEW

We need to model a newsletter dispatching system that:

1. Receives two kinds of streams: quick news and detailed articles.
2. Processes these streams using a text and image processing system.
3. Sends the processed feeds to three social media platforms: Facebook, Twitter and Instagram
4. The system should determine the average time required to publish and article over all its media.

In this project, the goal is to determine the average time it takes for the same article to be published on Facebook, Twitter, and Instagram.

Key Observations

Open multiclass model

The system in question is represented as an open multiclass queuing model due to the two separate streams (news and articles) with distinct behaviors coming from the external environment.

Text and image processing system: This is the initial stage where articles and news are prepared for publication.

Sending queues: Each social media platform (Facebook, Twitter, Instagram) has its own queue where processed articles wait to be published.

System Efficiency

We can clearly see that the system is efficient in handling quick news due to its higher arrival rate and lower processing time but detailed articles are less frequent, introduces higher processing demands which are exacerbated by the varying processing times on different social media platforms.

Percentages not summing to 100%

The percentages of articles or news items published on each social media platform do not sum to 100%. This is because each feed can be published on multiple platforms or may not be published at all.

P(facebook & instagram) = 0.8 x 0.9 x 0.6 = 0.432 or 43.2%

P(all socials) = 0.8 x 0.1 x 0.6 = 0.048 or 4.8%

P(not published) = 0.2 x 0.9 x 0.4 = 0.072 or 7.2%

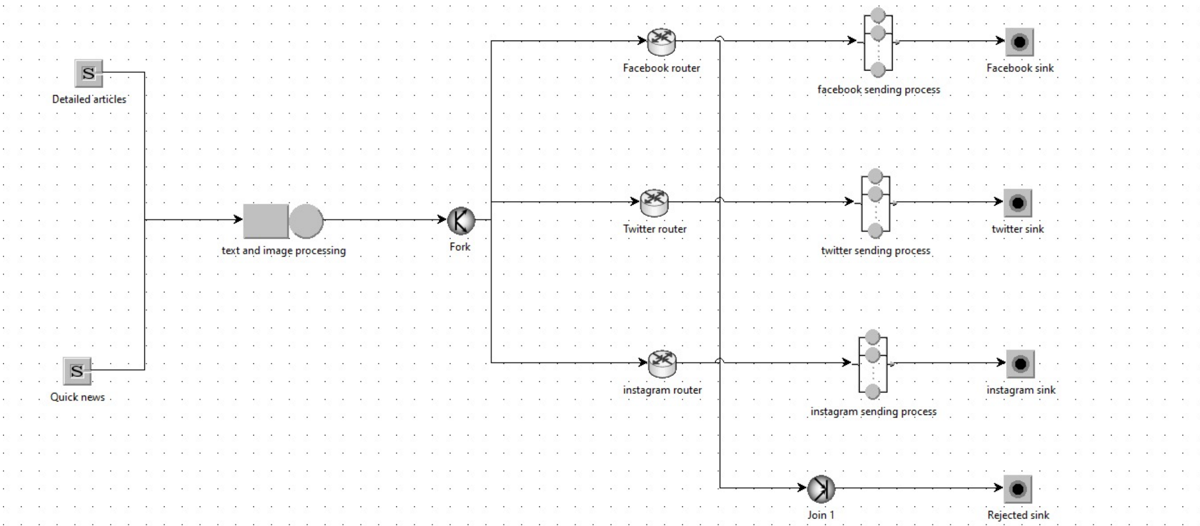
By calculating these probabilities, we can better understand how frequently an article is published across multiple social media platforms or not published at all, including the likelihood of it being published on all three.

Publication Dynamics

Probability analysis shows that most articles are published on Facebook and Instagram together (43.2%) while only a fraction (4.8%) get published on all three platforms.

Modelling the system with JMT

We created a model that captures the process of handling and publishing news and articles across multiple social media platforms, allowing for the analysis of the average time required for an article to be published on all platforms.



Model components

1. Sources:
    - Quick news source: Generates stream of quick news items
    - Detailed Articles Source: Generates a stream of detailed articles.

2. Queueing station: A station where both news and articles undergo text and image processing. This station handles the convergence of both streams.

3. Fork: A mechanism used to triplicate each news item and article, enabling them to be sent to multiple social media platforms.

4. Routers: Individual routers for each social media platform (Facebook, Twitter, Instagram). These routers direct the processed jobs either to the corresponding social media queue or to a discarded jobs sink if they are not to be published on that platform.

5. Delay stations: Three delay stations, one for each social media platform. These stations represent the infinite server models of the social media platforms, accounting for the time taken to publish the feeds on:

    - Facebook
    - Twitter
    - Instagram

6. Join: A component used to collect all jobs that were not directed to one or more social media platforms.

7. Sinks: There are sinks for each social media platform and for discarded news or articles. These sinks represent the final destinations of the jobs, whether they are published or discarded

This model effectively captures the process of handling and publishing news and articles across multiple social media platforms, allowing for the analysis of the average time required for an article to be published on all platforms.

Detailed explanation and analysis

Context and probability

As mentioned earlier, observing the system over a long period, the probability of an article being published on all three social media platforms is 4.8%. This translates to one such appearing approximately every 20.83 articles. Therefore we need average of 20.83 articles for a feed to appear on all social media platforms.

Arrival rate and Inter-Arrival time

The arrival rate ($\lambda_{\text{articles}}$) for articles entering the system is 0.1 articles per second. The inter-arrival time, which is the average time between the arrivals of consecutive articles, is the inverse of this rate:

Inter-arrival time = $\frac{1}{0.1}$ = 10 seconds

It means one article arrives every 10 seconds on average, we multiplied this by our target article 20.83.

But we need to consider total time from when an article arrives to when it gets published.So we must consider the path of our model which includes the processing time and there are some delays while publishing these articles we should consider those values also.

Average time from arrival to publication

To determine the total time for a target article from arrival to publication on all three platforms, consider the following components:

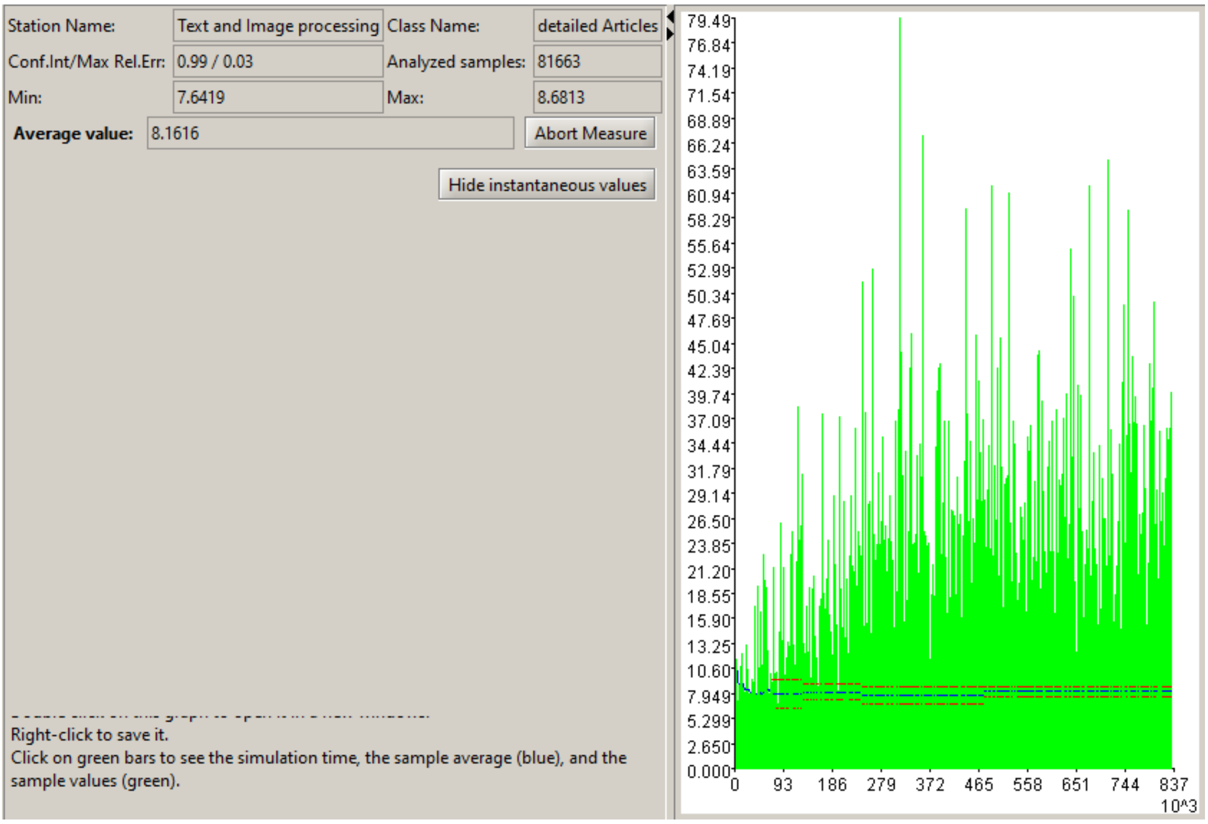
1. **Text and Image processing time:** The average response time of the text and image processing service center i.e. total time to process the article's text and images.

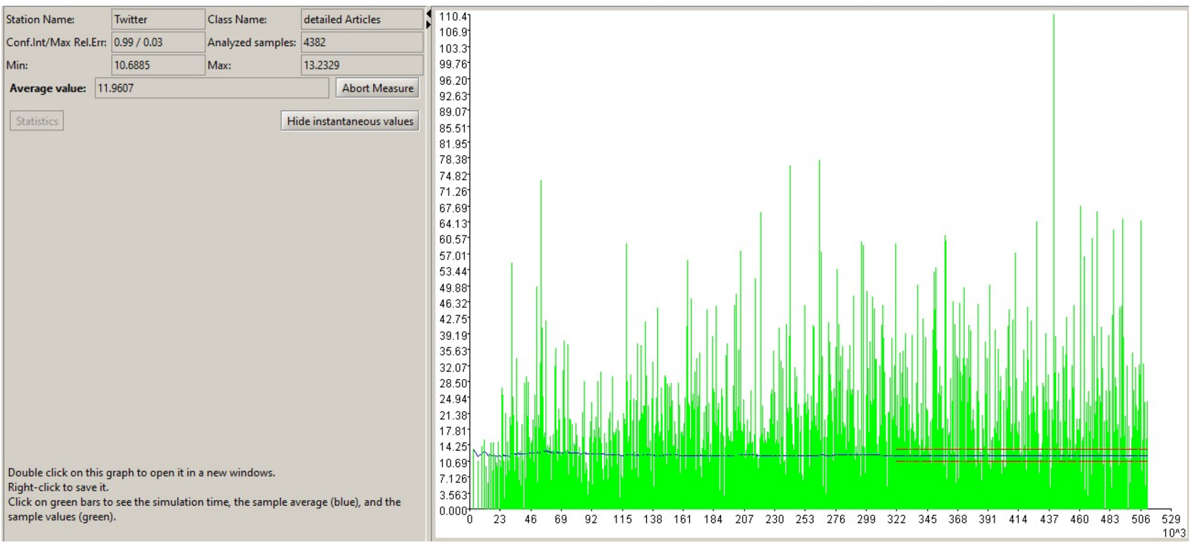
2. **Publishing delays**: Time taken to process the article's text and images.Each social media platform has an average processing time before publication:
   - Facebook: 10 seconds
   - Twitter: 12 seconds
   - Instagram: 9 seconds

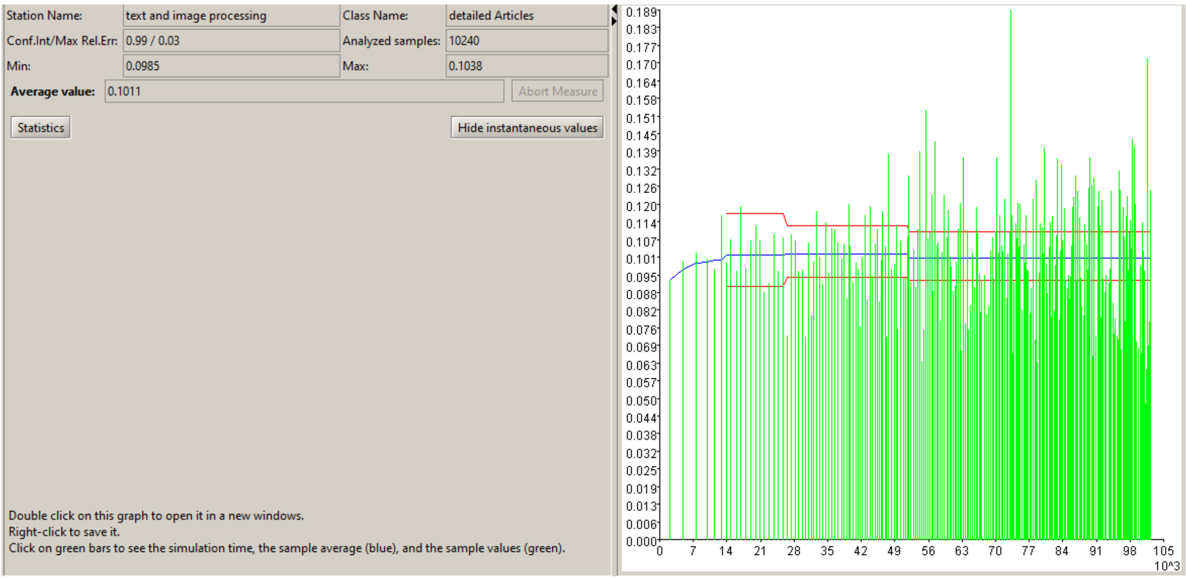
   Since the platforms are modeled as infinite server stations, the bottleneck for the publication process is the longest processing time, which is Twitter's 12 seconds.

Simulation for Text and Image processing

To estimate the total time the administrator should expect to wait, we sum the average entry time, the processing time, and the bottleneck delay. Now we will calculate the average response time for text and image processing and then we will add average entry time + bottleneck delay for both upper and lower bound.







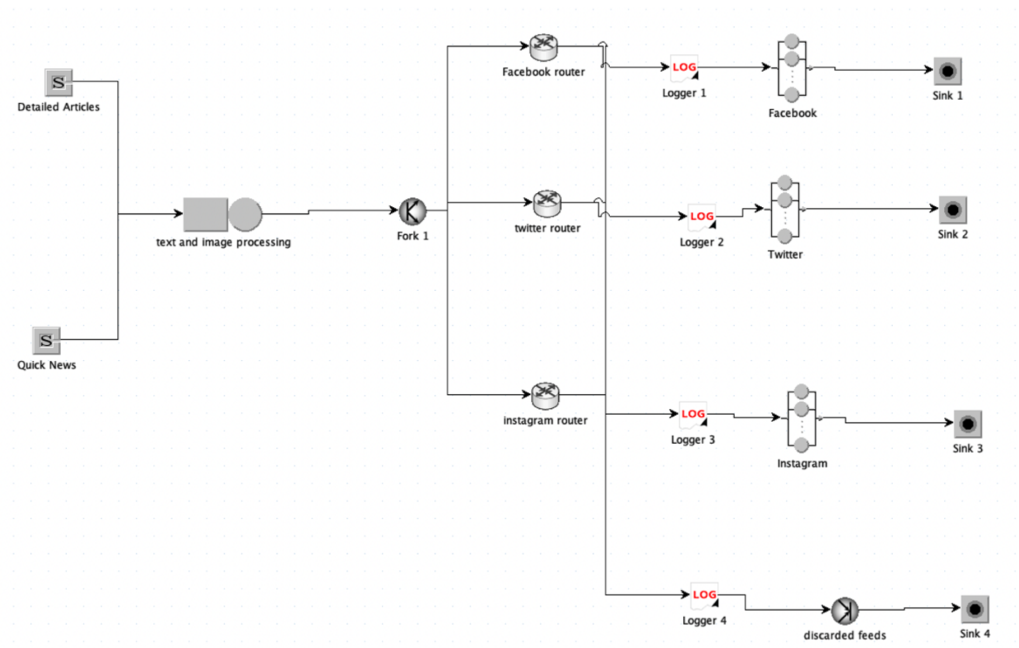
Therefore :

- Lower bound : (20.83 * 9.85) + 7.6419 + 10.6885 = 223.5059
- Upper bound : (20.83 * 10.38) + 8.6813 + 13.2329 = 238.1296

With 99% confidence, the administrator can expect that the time required for a target article to be published on all three platforms will fall between 223.5059 and 238.1296 seconds.

Verifying Accuracy Using JMT Logging Features

Now to verify the accuracy of our result we exploit the Logging Feature of JMT which is a tool to understand and debug the traffic flow moving through the interesting parts of the model. Logger were placed right after the routers to trace all the jobs passing through the system.



The following is the table returned by the logger:

	LOGGERNAME	TIMESTAMP	JOB_ID	CLASS_ID
1	Logger 4	1.219132	3	quick news
2	Logger 3	1.219132	4	quick news
3	Logger 2	1.219132	5	quick news
4	Logger 3	1.744439	7	quick news
5	Logger 4	1.744439	8	quick news
6	Logger 1	1.744439	9	quick news
7	Logger 4	2.721224	11	detailed Articles
8	Logger 1	2.721224	12	detailed Articles
9	Logger 3	2.721224	13	detailed Articles
10	Logger 3	3.153658	16	quick news
11	Logger 2	3.153658	17	quick news
12	Logger 1	3.153658	18	quick news
13	Logger 2	3.316929	21	quick news
14	Logger 3	3.316929	22	quick news
15	Logger 1	3.316929	23	quick news
16	Logger 3	3.373992	24	quick news

After obtaining the results from the simulator, we used R Studio to analyze the jobs sent to all three social media platforms. This involved applying specific filtering code to extract the necessary information, resulting in a new dataset that contained the arrival times of these jobs.

The next step in our analysis was to calculate the interarrival times. We did it by subtracting each arrival time from the one that followed it in the dataset. This process give us a set of interarrival times.

After conducting a simulation in JMT, the logging stations traced the traffic flow, resulting in the table previously presented. This table became the focus of the code. The outcome closely matches the results obtained from the earlier method.

```
>
> mean(Interarrival_time$Time)
[1] 208.7623
>
```

```
library(dplyr)

# Load the data
global1 <- read.csv('C:\\Users\\hp\\Desktop\\global.csv')

# Filter data for "detailed Articles" in CLASS_ID
global_article <- global1 %>% filter(CLASS_ID == "detailed Articles")

# Initialize an empty data frame for Arrival_Time
Arrival_Time <- data.frame()

# Process each unique timestamp
unique_timestamps <- unique(global_article$TIMESTAMP)

for (timestamp in unique_timestamps) {
  A <- global_article %>% filter(TIMESTAMP == timestamp)

  if (!"Logger 4" %in% A$LOGGERNAME) {
    Arrival_Time <- rbind(Arrival_Time, data.frame(Time = timestamp))
  }
}

# Calculate inter-arrival times
Interarrival_time <- data.frame()


for (i in 1:(nrow(Arrival_Time) - 1)) {
  interarrival <- Arrival_Time$Time[i + 1] - Arrival_Time$Time[i]
  Interarrival_time <- rbind(Interarrival_time, data.frame(Time = interarrival))
}


# Calculate and print statistical summaries
average_time <- mean(Interarrival_time$Time)
cat("Mean Inter-arrival Time:", average_time, "\n")
```

CONCLUSION

We evaluated the efficiency of our newsletter dispatching system and determined the average time needed to publish an detailed article across Twitter, Facebook and Instagram. With the help of JMT, we built a queuing network model of the system, incorporating parameters like arrival rates, delay stations and routers.

Both the methods gave us the same results. Therefore the time required for a target article to be published on all three platforms will fall between 223.5059 and 238.1296 seconds.