

Collecting Metrics from Emails

Vivek Kekuda T U*

*(Final Year Student, Department of Information Science and Engineering, R. V. College of Engineering, Bengaluru
Email: vivekkekuda29@gmail.com)

Abstract:

In this paper we discuss the various metrics that could be collected from emails that are sent out *en masse* to target users. We discuss not only the need to collect metrics, but also the kind of metrics can be collected and how to collect some of them. A design which scales well is proposed that could be used to capture these metrics individually for every target user part of an email campaign. The application should be built in a way that it can both send emails using SMTP servers as well as collect metrics from them.

Keywords —emails, metric collection, full stack development, gaining insights, email campaigns.

I. INTRODUCTION

There are hundreds of emails sent out to thousands of different users on a daily basis. Traditionally, one does not know how many of these emails were opened, or what interactions did the target user do with it. In the age that we live in, it becomes important to measure the reach of the emails that are sent out. By collecting metrics – like the number of emails sent out and the number of emails opened, one can calculate the reach of the emails sent out at an individual target user level. To make sure that the user interacts with emails, two things are of prime importance: the email subject line and the sender address [2]. User could delete an email which comes from someone outside of their contact list [8].

Full stack development is the process of writing both client and server-side software. The application, to collect metrics – must host the landing pages internally which the users land upon after clicking the links in the emails.

This paper describes how the metrics can be collected from emails. It goes over some of the important metrics that could be collected from real world email campaigns which are sent out to specific users. Using these metrics, the one sending out email campaigns – in this case the administrator

of the application, can gain insights and see the percentage of user interaction and further improve the reach of the emails.

The metrics collected from these email campaigns can be used in several different ways. One can be generating a dashboard to provide a quick overview of the results.

II. LITERATURE SURVEY

An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it.

A. ZirthingBawm Lian and Rudra Pratap Deb Nath

This paper [1] aims at aimed at analysing subscriber activity and presents some interesting results. It finds that improvements are made after sending out email campaigns to segmented groups of subscribers based on their individual interest. Finding this interest and personalising the emails to every user has resulted in better results.

B. Anurag Tiwari, Mohd. Aquib Ansari and Rachana Dubey

This paper [2] aims at finding the best optimal procedure to send the emails to users and get a high open rate. The success of an email campaign is

found to be dependent on two major factors, email sender address, and the email subject line.

C. S. Das, A. Kim, Z. Tingle and Christena Nippert-EngAll

This paper [3] presents a user focused study. It includes details about the target users, which is not often taken into consideration as part of email campaigns. It calls for better reporting and recruiting practices.

D. E. Fariborzi and M. Zahedifard

This paper [8] talks about the advantages, disadvantages and improvements that can be done to email marketing from an E-Commerce point of view.

III. METHODOLOGY

The methodology can be divided into two parts,

A. Understanding the need to collect metrics

In an application that can be used to send out emails to hundred thousand of users, it becomes important to also know how many of these emails are likeable and makes the user interact with it. A high open rate would mean that the user was interested in the email subject [2]. Similar conclusions can be drawn based on the kind of metrics that are collected. Analysis can be done on a target user if he clicked on a particular link, associations can be drawn on what kinds of emails grab the most attention, and so on. So, it becomes important to send trackable emails.

B. Identifying the metrics that can be collected

The important metrics that determines the success of an email that can be collected are:

- Total number of emails sent
- Total number of emails opened
- Total number of clicks done on the links present in the email
- Total number of emails reported and forwarded

These metrics further broken down for every user of a particular email campaign leads to determining –

- If the email was sent to him

- If he opened the email
- If he clicked links present in the email – if done, details on which link was clicked is also important, since an email may have multiple areas of interest.
- If he submitted data – if done, capture the submitted data which could be any form details.

All these metrics are nothing more than events that happen in the lifecycle of a campaign. Once a campaign is launched, emails are sent out to all the specified users and all events – email open, click link, submit data, etc. are captured and stored as part of the campaign results.

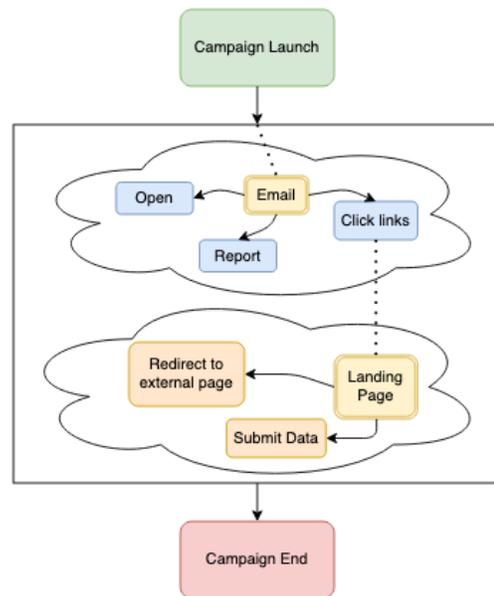


Fig. 1 Different events in an email campaign lifecycle

IV. IMPLEMENTATION AND RESULTS

A. Design proposal for metric collection

Since these metrics on a high level are events that happen in real time, a separate table to capture them is ideal. It can be linked back to a campaign and the target user using unique identifiers for campaigns and target users. A unique identifier for a user part of an email campaign would be his email address. As and when these events happen, the *events* table is

populated with unique attributes that link back to the campaign and target user, a *message* attribute to briefly describe said event, and an additional *details* attribute to further capture details about said event. The timestamp is also captured in a *time* attribute, which would be *CURRENT_TIMESTAMP*. All these attributes define every metric that is collected. It can be filtered based on the unique identifier of the campaign to get results for a campaign.

TABLE I
EVENTS TABLE SCHEMA

Field	Type	Extras
id	integer	Primary key, auto_increment
campaign_id	integer	-
email	varchar	-
time	datetime	current_timestamp
message	varchar	-
details	blob	-

B. Landing page considerations

The landing pages are also hosted from the same server that sends out mails using SMTP servers [10]. Each landing page URL must have a unique identifier which defines the user that clicked on it. A 4 character long randomly generated short string represents the target user. This is unique to every user that the email is sent out to, for a particular campaign. The landing page must also include details on the campaign under which the email was sent out. If the aim to collect data that is submit in the landing page, a form with an empty action and method as POST must be present – this allows the server to handle the POST event and capture the data that was submit. If redirecting to other external webpages, a redirect query parameter must be present, with it having the URL for the original webpage. In total, the URL landing page would look like,

`https://<hostname>/?cid=xx&uid=yyyy&redirect=xyz.com`

Query parameters:

cid – mandatory, denotes unique id of a campaign.
 uid – mandatory, denotes unique id of a user.
 redirect– optional, defines the page to be redirected to after capturing the click.

C. Capturing events

By handling the *GET* and *POST* methods on the landing page URL, one can capture the events that happen in real time. As and when the user clicks a link in the email, a GET request is placed to the landing page URL. Since our server would be listening to this, the clicked link event can be captured for the given user. The endpoint for the server to listen to would be,

`/{path:.*}`
 which listens to all combinations of the landing page URL.

If the method was GET, then there are two different scenarios.

1. Presence of a redirect query parameter: In this case, the user intends to go to an external webpage, but we need to capture the event before actually redirecting the user to it. The redirect query parameter is read and its value which denotes the original link is used to the redirection, after saving the details in a *Clicked External Link* event.
2. Absence of a redirect query parameter: In this case, the landing page is loaded, and the event *Clicked Link* is captured. It still has the campaign identifier and target user identifier as its query parameters.

If the method was POST, then the submitted data is sent along with the request in its body. We read all the values and save the details in a *Submitted Data* event.

The events table would start populating post campaign creation and some target user interaction.

TABLE III
MOCK DATA IN EVENTS TABLE

id	email	time	message

1	-	Hh:mm	Campaign Created
2	user1@email.com	Hh:mm	Email Sent
3	user2@email.com	Hh:mm	Email Sent
4	user1@email.com	Hh:mm	Clicked Link
5	user2@email.com	Hh:mm	Clicked External Link
6	user1@email.com	Hh:mm	Submitted Data
7	user2@email.com	Hh:mm	Clicked Link

This is a filtered query for a particular campaign with identifier xx. The query looks like,

```
SELECT id, email, time, message FROM events
WHERE campaign_id=xx;
```

V. CONCLUSIONS

This paper presents why there is a need to include metric collection in applications which offer sending emails to multiple different users. There is almost always a need to know how *well* the emails are doing, that is, if it can gain the attention of target users.

This paper also identifies some of the key metrics that could be collected from emails that are sent out. It discusses the events that occur in an email

campaign lifecycle and proposes a scalable design to collect metrics in the form of events.

This paper also shows how a few metrics – *Clicked Link*, *Clicked External Link* and *Submitted Data* can be collected in detail. The table schema and mock data that is collected is also shown. The results could be filtered or aggregated based on the needs. It could be visualized, be further processed to gain insights, etc.

REFERENCES

- [1] Zirthing Bawm Lian and Rudra Pratap Deb Nath, "A Conceptual Model for effective email marketing", December 2014. Available: IEEEExplore, DOI: 10.1109/ICCIETech.2014.7073103
- [2] Anurag Tiwari, Mohd. Aquib Ansari and Rachana Dubey, "An Effective Email Marketing using Optimized Email Cleaning Process", April 2018. Volume-6, Issue-4. Available: Research Gate, DOI: 10.26438/ijcse/v6i4.277285
- [3] S. Das, A. Kim, Z. Tingle and Christena Nippert-EngAll, "About Phishing: Exploring User Research through a Systematic Literature Review", August 2019. Available arXiv, arXiv:1908.05897
- [4] Smart Insight Email marketing engagement and response statistics 2018, The best email statistics sources to benchmark open and clickthrough rates for your email campaigns in your industry sector.
- [5] Educational SQL resources, Microsoft SQL Server Documentation. Accessed on: 5 May, 2021. [Online]. Available: <https://docs.microsoft.com/en-us/sql/sql-server/educational-sql-resources?view=sql-server-ver15>.
- [6] Mark McGranaghan, "Go by Example". Accessed on: 10 March, 2021. [Online] Available: <https://gobyexample.com/>.
- [7] WHERE (Transact-SQL), Microsoft SQL Server Documentation. Accessed on: 10 May 2021. [Online]. Available: <https://docs.microsoft.com/en-us/sql/t-sql/queries/where-transact-sql?view=sql-server-ver15>.
- [8] E. Fariborzi and M. Zahedifard, "E-mail Marketing: Advantages, Disadvantages and Improving Techniques", June 2012. Volume-2, No-3. Available: ijeeee, <http://www.ijeeee.org/Papers/116-CZ02024.pdf>
- [9] Claudio Guidi, Ivan Lanese, Manuel Mazzara and Fabrizio Montesi, "Microservices: A Language-Based Approach", September 2017. Available: Springer, https://doi.org/10.1007/978-3-319-67425-4_13
- [10] Documentation, Gophish. Accessed on 10 March 2021. [Online]. Available: <https://pkg.go.dev/github.com/gophish/gophish#section-documentation>.
- [11] Microservices and Containerized Applications.' Apress, Berkeley, CA,2020. https://doi.org/10.1007/978-1-4842-6216-0_5