

Large Scale Analytical Insights of Email

Communication Patterns.

by

Lakshminarayana Motamarri

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved July 2011 by the
Graduate Supervisory Committee:

Raghu Santanam, Co-Chair
Jieping Ye, Co-Chair
Hasan Davulcu

ARIZONA STATE UNIVERSITY

August 2011

ABSTRACT

This thesis research attempts to observe, measure and visualize the communication patterns among developers of an open source community and analyze how this can be inferred in terms of progress of that open source project. Here I attempted to analyze the Ubuntu open source project's email data (9 subproject log archives over a period of five years) and focused on drawing more precise metrics from different perspectives of the communication data. Also, I attempted to overcome the scalability issue by using Apache Pig libraries, which run on a MapReduce framework based Hadoop Cluster.

I described four metrics based on which I observed and analyzed the data and also presented the results which show the required patterns and anomalies to better understand and infer the communication. Also described the usage experience with Pig Latin (scripting language of Apache Pig Libraries) for this research and how they brought the feature of scalability, simplicity, and visibility in this data intensive research work.

These approaches are useful in project monitoring, to augment human observation and reporting, in social network analysis, to track individual contributions.

DEDICATION

This thesis is dedicated to my parents, who supported me, provided their love and who stood by me all throughout my life.

ACKNOWLEDGMENTS

I am sincerely thankful to Dr. Raghu Santanam, who has provided his valuable inputs and guided me throughout my research work.

I would like to thank Dr. Jieping Ye to accept to be in my Committee and for providing his valuable inputs.

I also thank Dr. Hasan Davalcu to accept to be in my committee.

I would like to acknowledge the names of my friends who are indirectly part of my success by providing their support. Vamsi Sripathi, Vishnu Sudha, Praveen Gorthy, Aditya Jupudi, Ujwal Koneru, Sashi Gangaraju, Divakar.

Finally I thank my family for their support, love and encouragement.

TABLE OF CONTENTS

	Page
LIST OF TABLES	vi
LIST OF FIGURES	vii
CHAPTER	
1 INTRODUCTION	1
1.1 Research Questions	2
2 LITERATURE REVIEW	6
3 DATA SOURCES	10
4 APACHE PIG AND HADOOP	11
4.1 Apache Pig	11
4.2 Pig Latin	12
4.3 Cloudera's CDH	14
4.4 Google/IBM Virtual Infrastructure support - Cluster	15
5 METHODOLOGY OF MEASUREMENTS	17
5.1 Question-1: Volume of Emails	17
5.2 Question-1 (b): Consistency among 5.1	20
5.3 Question-2: Active participants	21
5.4 Question-2 (b): Consistency among 5.3	26
5.5 Question-3: Buzzwords	28
5.6 Question-4: Mood of the Community	31
6 CONCLUSION AND FUTURE WORK	33
REFERENCES	34

LIST OF TABLES

Table		Page
1.	List of Ubuntu sub projects considered as data sets	10
2.	Xubuntu release dates	20
3.	Top 15 Active participants of first sub project	24
4.	Top 15 Active participants of second sub project	25
5.	Top 25 Buzzwords from second sub project	29

LIST OF FIGURES

Figure		Page
1.	Casual model representation of the factors considered for the metrics in observing the communication pattern	6
2.	Screen shot of the data loaded	13
3.	Screen shot of the data, which is grouped	13
4.	A screen shot of the Cloudera's CDH, being played using VMware player	14
5.	Total number of unique email conversations	19
6.	Total number of unique emails of a single project in conjunction with its release dates	21
7.	A tree map based visualization of active participants of 'Ubuntu-studio' project's mailing list	24
8.	A tree map of top 15 active participants of "ubuntu-devel-discuss" project's mailing list	25
9.	Screen shots of the Motion Chart with flow of Activity of two users ..	28
10.	Top 25 buzzwords from second sub project	30
11.	Graph with 15 buzzwords from "ubunutu-devel" sub project	31
12.	Top 10 moods of sub-project "ubuntu-studio-devel"	32
13.	132 Moods considered for the classification	33
14.	Measures considered for each Mood level	34
15.	Top level-1 Moods with high measurements	34

CHAPTER 1

INTRODUCTION

The open source software products have a significant share in their respective markets. The peculiarity of these open source projects is that they are not driven by the factor of money, like regular proprietary software. Here developers contribute just based on their passion and work satisfaction. They tend to choose the problem that they wish to work on and up on which they are more confident. This has created a lot of interest among researchers and also a case of necessity for the open source project's program managers, to keep a watch on the progress of these open source communities and make crucial decisions to mitigate risks. Also this analysis is important to already contributing developers to get a sense of their community and the buzz among its participants which could gain further confidence among existing and also motivate new users / developers to join.

The role that communication plays in a collaborative environment is major. Communication in any project is important, among the team members and in between various teams. So I decided to use this communication data in achieving our goal just by observing the patterns. Among various communication tools, Email has a major share. Be it major decisions or a simple bug report, email has become a de facto standard for communication among various industries. Coming to the case of open source projects, as these are distributed teams working virtually, I think email grabs a large share for communications as, they don't physically interact and online chatting application are also not used, as one cannot expect all to work together at the same time. So I decided to use

email log archives in analyzing the communication patterns and observe the data from different dimensions, to draw suitable inferences / conclusions from them.

In this study I presented four measures based on reviews of previous literature works and also based on analysis and enhancements to individual opinions gathered from these works. The goal is to observe the group communication over the course of the Ubuntu, an open source operating system's project lifetime from its email log archives. I examined the following factors as possible success predictors:

a) Volume of communication Information (Number of emails).

b) Active Communicating participants.

c) Examining the content of the messages (actual text of the Messages).

d) Frequency / Consistency among all above.

For each factor, I came up with results that address certain key questions.

The corresponding questions are presented below.

1.1 Research Questions

These questions are aimed at understanding the basic parameters of the communications (Emails).

Q1: What is the total number of unique email discussions that were generated in each of mailing lists considered?

I wanted to measure how big the volume of entire mailing lists is, in terms of unique mail conversations started by a participant of that mailing group.

Q1 (b): what is the number of emails per month of a single sub project that developers are involved with participants and check this pattern in conjunction with that project's Release Dates.

This will help us visualize the consistency among the email conversations among the participants of that project. I attempted to observe, if there exists any Saw Tooth patterns; i.e. a steep rise in the number of conversations at the time of project releases, which might show the in-consistency of the work.

Q2: Who are the active participants among these mail conversations?

This will help us identify active participants and consistent contributors from their contribution Index. Previous literature reviews show that a significant number of active users show's the strength / activeness of that community.

Q2 (b): What is the rate of Change in the contribution of the Active users?

Here again everything is broken down based on the timeline and from a Motion chart, I attempted to see how the activity of active users changes over the time.

Q3: What are the frequently used keywords in the emails?

This is to examine the content of the actual message of the email, from the log file. Here I can find the frequently used keywords or simply the Buzzwords of the community, by creating a Cloud of keywords.

Apart from above I would like to also propose three additional questions, of which I am working on one at present and other 2 can be considered as a future work:

Q4: What is the Mood of the community?

At present I am working on finding the Mood of the community from its email messages, by text classification. In the earlier question I have found the buzzwords. Similarly I can find the occurrences of few moods, which can be classified as: anger, happy, sad, working, etc. This way I am attempting to measure the Mood of the entire community from its emails and based on this measurement we can decide the Mood.

Q5: What are the countries that the participants are from?

This is an attempt to see, the scale of the global community that is using this product. This can be achieved by extracting the IP information from the logs and then visualize using a Heat Map.

Q6: Try to identify important communications?

This is an attempt to measure the importance of an email by its thread length (number of replies with same subject name), and then create clusters grouping similar one's together. This way I can visualize the section of important / significant emails, from corpse.

The reminder of this report is as follow. Section 2 present a literature review of few related research works, from which few measurements were considered and others which were enhanced / modified to fit the data set. Section 3 is about the description of the data set. Detail explanation about email log files and the different sub projects of Ubuntu, from which I considered these archives. Chapter 4 is about Apache Pig libraries and Hadoop cluster. It explains about Pig Latin scripting language of Apache Pig library with an example. Section 5 is all about implementations of the Measurements and their Methodologies. Includes most of

my code and the information about the data processing I approached in materializing these metrics on the data set. Then, finally I concluded with suggestions for future work.

CHAPTER 2
LITERATURE REVIEW

Part of this research in measuring the progress of an OSS project I did a literature review on existing research work. Based on the review, I used slight variations of existing measures to fit my data set and also framed some entirely new measures.

Edith Anderson et al. (Edith Anderson, Annette Bergman and Lars Hallen) suggested the dimensions of the communication patterns that can be tracked. They are, i) Who are the communicators, ii) What is the information being communicated and iii) What is the frequency of this communication? Along with these for better illustration I added another dimension, which is the Volume (Quantity) of the information. The number of unique email messages of that mailing list.

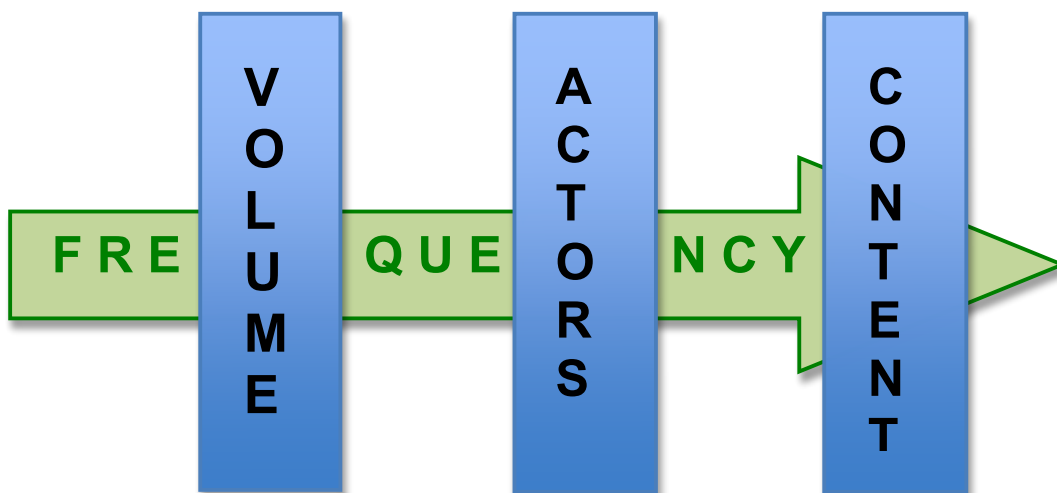


Figure. Casual model representation of the factors considered for the metrics in observing the communication pattern.

And more importantly I used the time factor (the frequency of the communications) among all the other three, which I felt would give a deeper visibility to each of the other dimensions or perspectives. The above diagram gives a Conceptual view of the description so far presented.

Dindin Wahyudin et al. (Dindin Wahyudin and A Min Tjoa) [1] has proposed the importance of measuring the average number of emails per month for an event based monitoring system of an OSS project. According to them, this event based monitoring application would benefit a manager to visualize the entire project development and also helps him to augment the human reporting.

Pater A. Gloor et al. (Peter A. Gloor, Rob Laubacher, Scott B.C. Dynes, Yan Zhao) [2] proposed a metric of identifying people that played a significant role in success of a COIN (Collaborative Innovation Networks). They analyzed the individual communication behavior by calculating everyone's simple contribution index: $\{\text{messages sent} - \text{messages received}\} / \{\text{total messages sent \& received}\}$. I used a variation of this equation in finding the active participants among the Ubuntu community. The variations are required so that it suits the data set. The reason is due to the email log, which has the recipient as a group email-ID and not an individual email-ID. The details are explained in detail in section 5.3.

H. Keith Edwards [3] presented the idea of observing and measuring the consistency of messages over the course of the project. This would give an abstract idea of the uniformity that is followed among the participants of the community. To check if there exist any SAW Tooth patterns corresponding to the project release dates.

I also followed this principle of consistency for Active participants, which would show the consistency among their contributions as well. I attempted to visualize this with a Motion gadget, which shows the flow of the activity and can help us identify any anomalies.

Dindin Wahyudin et al. (Dindin Wahyudin and A Min Tjoa) [1] also presented the scalability as one of the challenges in monitoring a distributed software development process. This is obvious in the present phase of Computers, where controlling and limiting data is inevitable. This is one reason why I choose to work on a Hadoop cluster. Hadoop has few higher-level interfaces, which make the work easy in carrying out the task and communicate with the cluster. One of them is Apache PIG. It has a scripting language called Pig Latin, which I used in implementing my measures.

Fazel Keshtkar et al. (Fazel Keshtkar and Diana Inkpen) [12] presented one of the methods of mood classifications for blog posts. They have identified 132 moods and classified them into 5 hierarchies / levels and have analyzed various levels of classifications. I am attempting to find the frequencies of few selected moods and try to understand the mood of the community.

Apart from all these, as part of future agenda I would like to propose an additional feature, as mentioned earlier in the Introduction. I.e. retrieving the IP address information. This can be tracked and can be visualized as a heat map on various places, which would give the layout of the countries and people that are using the product. This would help one to understand the reach out of the

product. And once this is viewed on a timeline basis, one can see the increase or decrease in the usage of the product by a global market.

CHAPTER 3
DATA SOURCES

The data source I considered here is Ubuntu’s email log archives. These are accessible at <https://lists.ubuntu.com/>. These are categorized into many sections like Development lists, Quality Assurance lists, bug lists etc., out of which, I have considered only development lists, which has the developers mailing activity among themselves and with users. {Note: No information is available to identify and categorize developers and users from them}. This has been further categorized based on sub projects like, ubuntu-development, kubuntu-development, kernel-team, ubuntu-mozilla- team, xubuntu development.

These archives are around 5 to 6 years old, ranging between 2004 to present.

Below are the sub-groups whose mailing archives, which I have considered.

(Lists with a minimum of 10MB size are considered).

Name of the sub project	Size of the log file	Start date	End date	Number of lines the log file has.
a) Ubuntu Development List	126MB	Sep 2004	Jun 2011	2,940,110
b) Ubuntu Development discuss List	59.2MB	Dec 2006	Feb 2011	1,265,023
c) KUbuntu Development List	21.2MB	Jan 2005	Jun 2011	500,252
d) Ubuntu Desktop List	10.8MB	May 2005	Feb 2011	256,533
e) Ubuntu-motu List	34.1MB	Nov 2005	Feb 2011	777,725
f) Ubuntu Mozille team List	11.5MB	Jan 2007	Oct 2010	213,846
g) Ubuntu Server List	26.9MB	Dec 2005	Feb 2011	591,459
h) Ubuntu Studio development List	10.9MB	May 2007	May 2011	402,443
i) XUbuntu Development List	39.1MB	Sep 2005	Feb 2011	824,349

Table: List of Ubuntu sub projects considered as data sets

CHAPTER 4

APACHE PIG AND HADOOP

With increasing amount of email log files and even for that matter any web data, the problem of scalability is obvious in present generation. In the recent past MapReduce paradigm based Hadoop has been largely used in data intensive applications. And few of the high-level abstractions written over Hadoop like Apache Pig, Cassandra and Hive, etc., also got into industry and are being used for large-scale data analysis. One of the query languages of Apache Pig is Pig Latin.

4.1 Apache Pig

I can use Apache Pig libraries and query with Pig Latin scripts on different kinds of data [10]. Pig supports unstructured data, like natural language text or just a log file with no schema. Our data set is an unstructured log file, which has no schema. An example of this log file is presented in Chapter 4. Pig also supports relational and nested types of data models.

As any other open source project or any other Apache incubated project pig takes in new code from its users. I can always add in UDFs – User Defined Functions into our data processing. In general, these UDFs are written in Java programming language. These can also be written in any other scripting language, like Jython that can be compiled down to Java.

Pig's command line interface is called as a Grunt.

In MapReduce mode the option to achieve parallelism is by using PARALLEL clause.

The Pig compiler has an optimizer which optimizes / rearranges the flow of operations written in scripts. I can also make Pig not to do any optimizations, by just turning it off.

The Pig compiler transforms the Pig Latin scripts into Hadoop statements, which are styled in MapReduce. The important point to note here is in general Pig Latin scripts are not fashioned as MapReduce, though I can also express a MapReduce program in Pig Latin.

4.2 Pig Latin:

Pig Latin is a scripting language developed by Yahoo! Research. Pig Latin scripts are a sequence of steps, which help us carry out data analysis. This fashion of sequence of steps in programming a task is very much similar to any other programming language and hence appealing to programmers. Its simplicity and cleaner approach and the way one can easily visualize the query execution flow is also attracting non-programmers like – physicists, economists, oceanographers, etc, many of whom deal with large amounts of data.

All the Pig Latin scripts that are developed for this research work are explained in next section. For a quick reference on understanding the basics of Pig Latin and more precisely understanding the difference between the control flow in Column based MapReduce and Row based SQL approaches, I present following

Example: To count number of lines in given text file.

```
Loaded_File = LOAD 'example.txt';
```

```
Counting_File = FOREACH Loaded_File GENERATE COUNT (Loaded_File);
```

This throws you an error. The general flow in a regular programming paradigm is that I tend to count the rows. Here I need to group initially all the rows, into a

single column. Then I can use the COUNT method, which will count the number of items in that TUPLE, which is formed by the GROUP method. TUPLE is data structure, which is available in Pig.

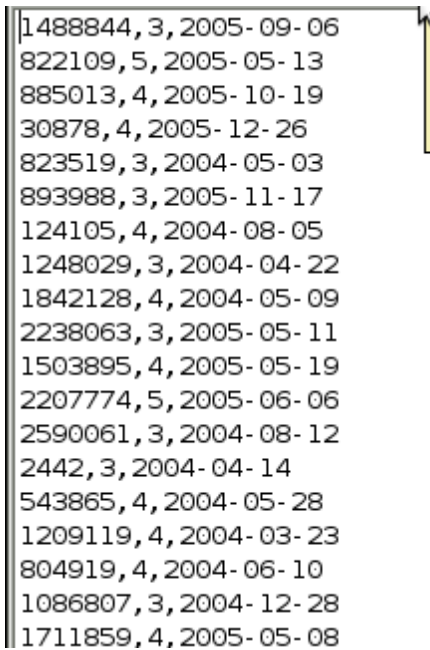
Example: {Here, is an example for TUPLE}

```
Loaded_File = LOAD 'example.txt';
```

```
Grouping_File = GROUP Loaded_File ALL;
```

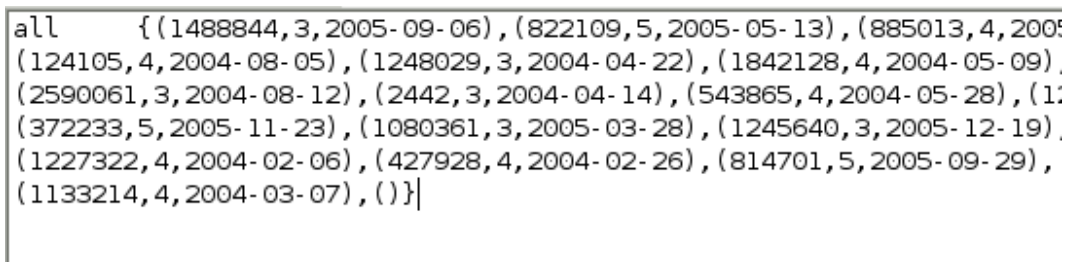
```
Counting_File = FOREACH Grouping_File GENERATE COUNT (Loaded_File);
```

Example data, which is loaded: *Figure: Screen shot of the data loaded*



```
1488844,3,2005-09-06
822109,5,2005-05-13
885013,4,2005-10-19
30878,4,2005-12-26
823519,3,2004-05-03
893988,3,2005-11-17
124105,4,2004-08-05
1248029,3,2004-04-22
1842128,4,2004-05-09
2238063,3,2005-05-11
1503895,4,2005-05-19
2207774,5,2005-06-06
2590061,3,2004-08-12
2442,3,2004-04-14
543865,4,2004-05-28
1209119,4,2004-03-23
804919,4,2004-06-10
1086807,3,2004-12-28
1711859,4,2005-05-08
```

Grouped data is below: *Figure: Screen shot of the data, which is grouped.*



```
all { (1488844,3,2005-09-06), (822109,5,2005-05-13), (885013,4,2005-10-19), (124105,4,2004-08-05), (1248029,3,2004-04-22), (1842128,4,2004-05-09), (2590061,3,2004-08-12), (2442,3,2004-04-14), (543865,4,2004-05-28), (1209119,4,2004-03-23), (804919,4,2004-06-10), (1086807,3,2004-12-28), (1711859,4,2005-05-08), (372233,5,2005-11-23), (1080361,3,2005-03-28), (1245640,3,2005-12-19), (1227322,4,2004-02-06), (427928,4,2004-02-26), (814701,5,2005-09-29), (1133214,4,2004-03-07), () }
```

In above figure, (1488844,3,2005-09-06) is an example of a TUPLE.

And $((1488844,3,2005-09-06), \{822109,5,2005-05-13\}, \dots)$ is an example of a data structure called BAG. Pig can count the number of TUPLES in that BAG, and assign the same to the ID “all”.

4.3 Cloudera's CDH:

Cloudera (<http://www.cloudera.com/>) is a Hadoop based software and services providing company. It releases a customized edition of a virtual machine with Ubuntu Operating system and pre configured Hadoop and Pig in them. I used this CDH Cloudera's distribution of Hadoop for this research work, from which I configured to the cluster sponsored by Google/IBM Virtual Infrastructure support (<https://univsupport.hipods.ihost.com/>).

I used Cloudera's CDH 2.0, which has Ubuntu 8.10 version, Hadoop 0.20 version and Pig 0.4.99 version.



Figure. A screen shot of the Cloudera's CDH, being played using VMware player.

4.4 Google/IBM Virtual Infrastructure support:

Google and IBM sponsored research cluster is of 530 Nodes with HDFS of 350750.7 GB / 594012.2 GB and JobTracker, the main processing agent which keeps track of all processes / jobs is of 1 / 2634 Maps and 1 / 878 Reduces. The version of Hadoop used in the cluster is 0.20.1

One can access the cluster using a SOCKS proxy and by configuring the local Pig client (e.g. Cloudera's Virtual Machine - CDH) with the below mentioned configurations. The Pig configuration files need to be configured to point to the cluster that you are using.

Note: Access to this Google/IBM cluster at present is only through reference.

Here following configurations are made:

Core-site.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="****.xsl"?>
<configuration>
<property>
    <name>fs.default.name</name>
    <value>hdf://*****com:****</value>
    <description> ... </description>
</property>
.....
<property>
    <name>hadoop.job.ugi</name>
    <value>Username,group-name</value>
</property>
<property>
    <name>hadoop.tmp.dir</name>
    <value>/tmp/hadoop-username</value>
    <description>.....</description>
</property>
```

```
</configuration>
```

Similarly Mapred-site.xml also to be configured:

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="***.xsl"?>
<configuration>
<property>
    <name>mapred.system.dir</name>
    <value>/tmp/hadoop/.../</value>
    <final>>true</final>
    <description>... </description>
</property>
<property>
    <name>mapred.job.tracker</name>
    <value>99.99.99.99:9999</value>
    <description>...</description>
</property>
</configuration>
```

CHAPTER 5

METHODOLOGY OF MEASUREMENTS

This chapter has all the explanations about the questions and their results for which the data is processed and measured.

5.1 Question-1:

What is the total number of unique email discussions that were generated in each of mailing lists considered?

First thing before I move forward I shall present you what all information is stored in the Email logs and which part of them are extracted.

Example of part of a log file:

```
From soifran@piem.org Sat May 12 11:59:07 2007
Received: from [81.201.185.232] (helo=kiwi.servaux.org)
Subject: Orca Fails in Studio.
To: ubuntu-studio-users@lists.ubuntu.com
Content-Type: text/plain
Date: Sat, 12 May 2007 12:56:22 +0200
Hi everybody
```

.....

The important part here is how to identify of each unique email, or simply choose the keyword or combination of keywords to identify an unique email? The possible combinations could be: a line with the following combination of keywords:

- i) 'From' + '@' - 'From:'
- Or, ii) 'Subject'
- Or, iii) 'From:' and '@' and '.'

Once I parse the log for above combination of keywords and count the number of lines the result would give the total number of emails for that data set.

Pig Latin script for 'From' and '@' combination of keywords:

```
A = LOAD 'ubuntu-studio-devel.mbox'; // You load the log file.
B = FILTER A BY $0 MATCHES '.*From.*' and NOT $0 MATCHES '.*From.:.*';
C = FILTER B BY $0 MATCHES '.*@.*'; // extract lines having '@' symbol
D = DISTINCT C; //further extract only unique lines from above.
// Below you extract lines separately on basis of Month.
E = FILTER D BY $0 MATCHES '.*Jan.*';
STORE E INTO 'Jan.txt';
E = FILTER D BY $0 MATCHES '.*Feb.*';
STORE E INTO 'Feb.txt';
E = FILTER D BY $0 MATCHES '.*Mar.*';
STORE E INTO 'Mar.txt';
...
```

Now once I have extracted all the files, which are grouped by month, I need to check the number of lines of each files, which it self is the number of emails for that corresponding month. This can be done with another Pig Latin script.

Pig Latin script for finding the frequencies (count) of above outputs:

```
A = LOAD 'Jan.txt';
B = GROUP A ALL;
X1 = FOREACH B GENERATE COUNT(A);
A = LOAD 'Feb.txt';
B = GROUP A ALL;
X2 = FOREACH B GENERATE COUNT(A);
...
A = LOAD 'Dec.txt';
B = GROUP A ALL;
X12 = FOREACH B GENERATE COUNT(A);
DUMP X1; DUMP X2; DUMP X3; DUMP X4;
```

DUMP X5; DUMP X6; DUMP X7; DUMP X8;
DUMP X9; DUMP X10; DUMP X11; DUMP X12;

The results of this script is shown in a graph below

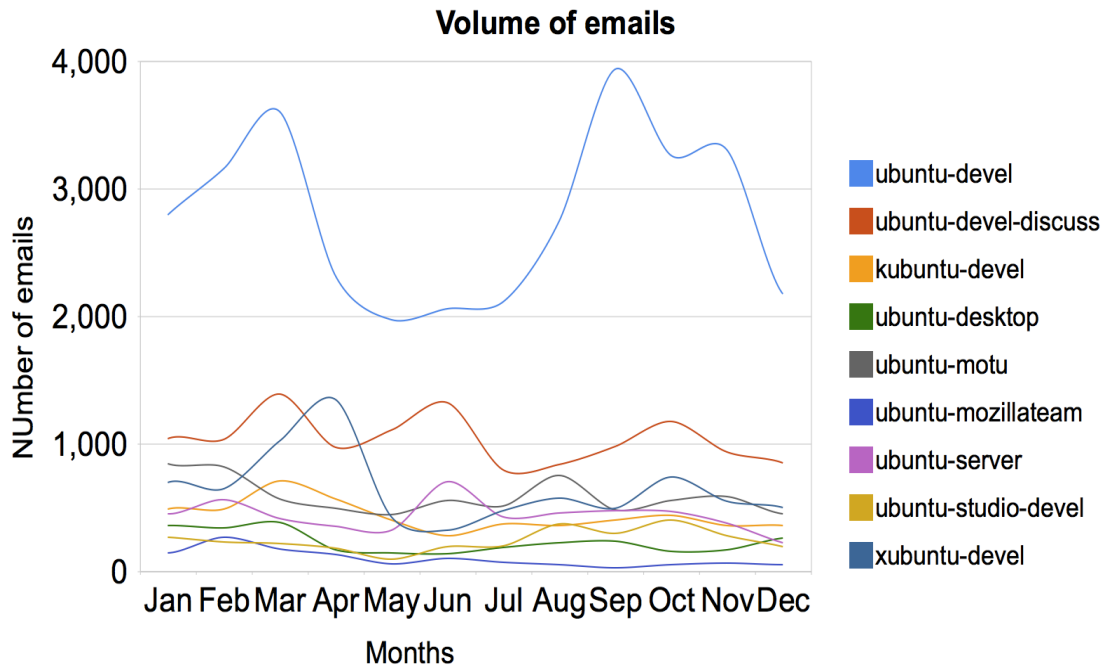


Figure: Total number of unique email conversations – categorized by month.

On average each month seems to have around 300 emails. And this data is collected between the year 2005 and 2011, which is 6 years of time period. So on average each sub project has around 50 ($300 / 6$) unique email conversations per month in those 6 years of margin.

This way one can visualize the Volume of emails, and see the average performance in terms of generating unique email conversations.

One typical insight from above graph can be raised by a question: why there is a huge bump for the “xubuntu” project in the month of April. So the reason can be observed when we check the release dates of the xubuntu project:

Version	Codename	Release date
5.10	Breezy badger	13 October 2005
6.06	Drapper Drake	June 2009
6.10	Edgy Eft	25 April 2008
7.04	Fiesty Fawn	19 October 2008
7.10	Gusty Gibbson	18 April 2009
8.04	Hardy Heron	April 2011
8.10	Interpid Ibex	April 210
9.04	Jaunty Jackalope	October 2010
9.10	Karmic Koala	April 2011

Table: Xubuntu release dates

Source: <http://xubuntu.software.informer.com/wiki/#Releases>

This observation shows the necessity to check these email conversations in accordance with their release dates. Hence the next measure is about calculating all email conversations based on their timeline.

5.2 Question-1 (b):

What is the number of emails per month of a single sub project that developers are involved with participants and check this pattern in conjunction with the project's new versions release dates.

This will help us see the consistency of the email conversations and see if this has been influenced by Releases. The script is very much similar to above script. The difference is that it is in specific to a single sub project and the graph is included along with its release dates.

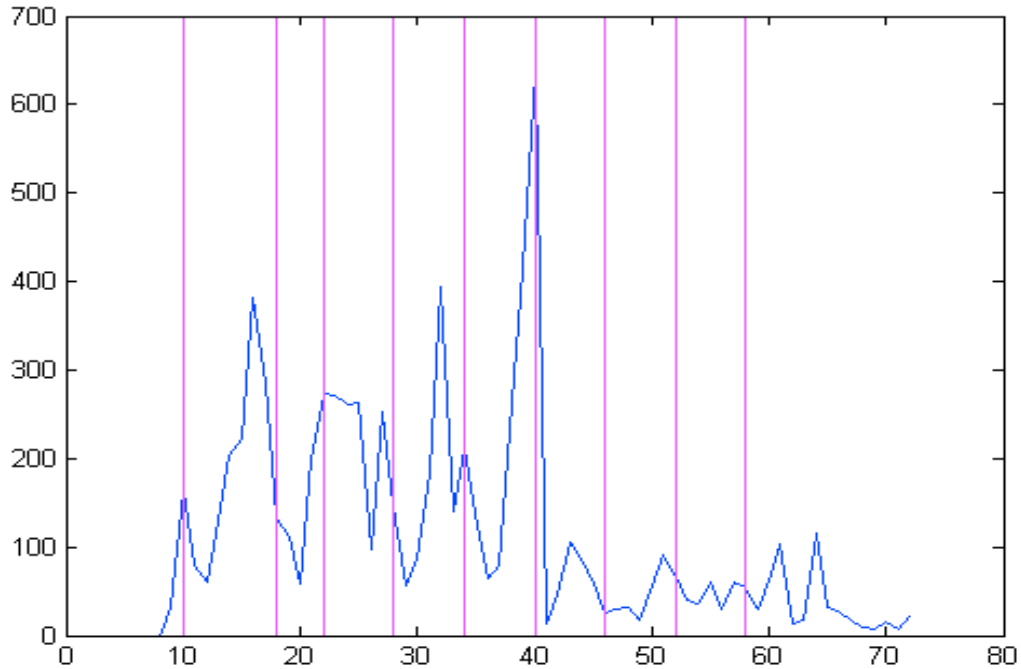


Figure: Total number of unique emails of a single project in conjunction with its release dates.

This way one can see if there are any “SAW Tooth” patterns among the communications.

5.3 Question-2:

Who are the active participants among these mail conversations?

This will help us identify active participants and contributors from their contribution Index. Literature reviews show that a significant number of active users show's the strength / activeness of a community.

Here the contribution index of each user is calculated from their participation.

One interesting factor is: As the data set is an email archive, which belongs to various groups, the recipient mail ID is always a group ID, and not a single user ID. Hence I exactly don't know in specific, how many times each user was involved (participated, not initiated) in those group conversations.

- ❖ One option is to consider, *Number of emails participations is equal to Number of total group's email conversations*. Because a user who is subscribed to a group ID, will be receiving an email to every email posted to that group. This would be a case, when I assume that all the users are in fact subscribed to that group since 2005.
- ❖ Also there could be another conflict, where an user would have been newly subscribed to that group ID, and even for such users, I am considering them as subscribed since 2005.
- ❖ There are few cases where users reply to the emails of initiators. I.e. few instances of "To:" & "Reply-To:" exist. I haven't considered these for now.

Hence based on above conflicts, I propose to give less importance to 'Num of email participants', i.e. by factorizing it with 0.5.

Contribution index =

$$\frac{\{ \text{Num of email initiations by user} + \text{Num of email participations by user} * \text{factor} \}}{\{ \text{Num of total group's email conversations} \}}$$

I.e. it is a Sum of number of times a specific user started the conversation and the number of times he was involved or participated, factorized by say, 0.5. This whole is divided by total number of email conversations that went in that sub group, during the entire period of that project.

Fig Latin script for finding Active participants from sub projects:

```
A = LOAD '16ubuntu-studio-devel.mbox';
```

```
B = FILTER A BY $0 MATCHES '.*From:.*';
```

```
// Below will tokenize all the sentences into words and place
```

```
// each word in a new line.  
  
C = FOREACH B GENERATE FLATTEN (TOKENIZE ($0));  
  
D = GROUP C BY $0;  
  
E = FOREACH D GENERATE group, COUNT(C);  
  
F = FILTER E BY $0 MATCHES '.*@.*';  
  
G = ORDER F BY $1;  
  
H1= FOREACH G GENERATE $0; // separate first column  
  
H2 = FOREACH G GENERATE $1; // separate second column
```

The step H1 will give you a set of all email ID, which is an identification of a user.

Note-1: I manually identified any non-user email IDs, which doesn't belong to a normal user. For example email-ID of a bug-reporting group like: <260558@bugs.launchpad.net>. I observed that for sub project "ubuntu-studio-devel" around 0.054 % of above-mentioned noise data are observed.

Note-2: Right now I have not resolved cases like: cjwatson@ubuntu.com and cjwatson@flatline.org.uk. Here there is a major probability that both emails might belong to a single user.

Note-3: In most of the cases I observed the occurrences of <scott@ubuntu.com> and scott@ubuntu.com. One way I can simply use a REPLACE method from Pig's string operations. Right now the results are manually done, to resolve this exception. I.e. I manually deleted all the occurrences of < and > symbols, in between the script execution.



Figure: A tree map based visualization of active participants of 'Ubuntu-studio' project's mailing list.

Note: This above tree map is formed without the contribution index. I.e. it is simply based on Number of email initiations by that user.

Top 15 active participants from 'Ubuntu-devel' group are:

Email ID	Initiations	Total group mails
mdz@ubuntu.com	3847	68363
dlist@ubuntuforums.org	1202	68363
glud-ubuntu-devel@m.gmane.org	1141	68363
ubuntu@kitterman.com	986	68363
mdz@canonical.com	949	68363
dennis@kaarsemaker.net	874	68363
daniel.holbach@ubuntu.com	843	68363
martin@piware.de	797	68363
martin.pitt@ubuntu.com	759	68363
cjwatson@ubuntu.com	659	68363
apache@volcano.xlogicgroup.com	653	68363
cjwatson@flatline.org.uk	578	68363
www@wailuku.xlogicgroup.com	519	68363
mjg59@codon.org.uk	383	68363
mjg59@srcf.ucam.org	321	68363

Table: Top 15 Active participants of first sub project.

This shows the active participants among 1st sub project.

Similarly for the 2nd sub project, the corresponding active participants are:

macoafi@gmail.com	882	25357
ciancia@di.unipi.it	754	25357
ubuntu@kitterman.com	539	25357
gludd-ubuntu-devel-discuss@m.gmane.org	402	25357
lists@janc.be	375	25357
ubuntu@bugabundo.net	342	25357
eapache@gmail.com	323	25357
mah@jump-ing.de	319	25357
mpt@canonical.com	293	25357
doctormo@gmail.com	257	25357
christopher.chan@bradbury.edu.hk	248	25357
remco47@gmail.com	228	25357
shirishag75@gmail.com	218	25357
andrew-ubuntu-devel@pileofstuff.org	213	25357
mdz@ubuntu.com	213	25357

Table: Top 15 Active participants of second sub project.



Figure: A tree map of top 15 active participants from “ubuntu-devel-discuss” who are listed in above table. Note: This above tree map is formed without the contribution index. I.e. it is simply based on Number of email initiations by that user.

To visualize more specific data we can observe the change of activity of specific users. This can be visualized by a motion chart and is explained in next section.

5.4 Question-2 (b):

What is the rate of Change in the contribution of the Active users?

Here I try to observe how the contribution of active users changes over the time period. For this I considered, the 2nd sub project – “Ubuntu-devel-discuss”, whose top 15 active users are displayed in above table. Here I observe all the activities of these users, by breaking down their contribution in to years and months.

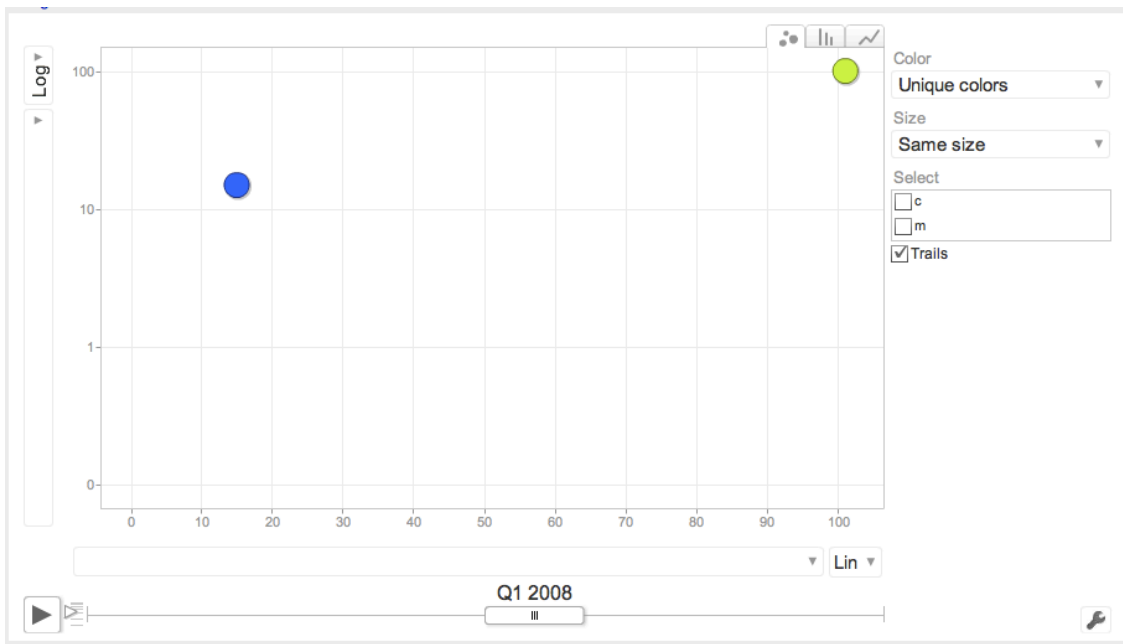
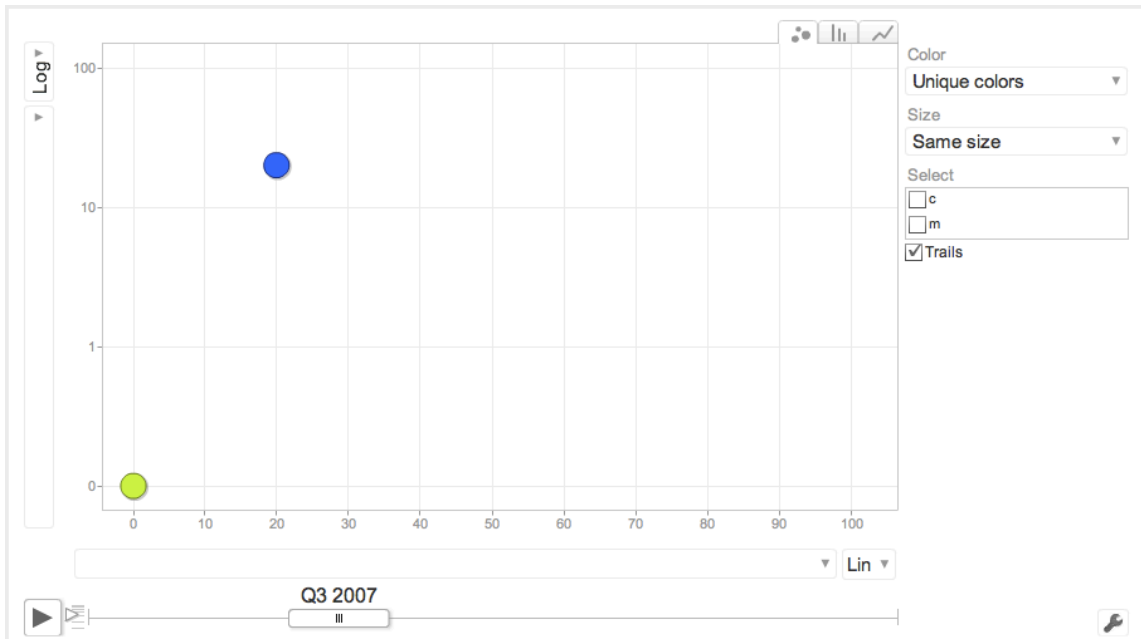
This feature of measuring the consistency of the activeness of a user, is inspired from question 1(b), where I measured the frequency of the mailing lists which are done based on a time line. Similar I assigned the same concept to question 2, the consistency of active users.

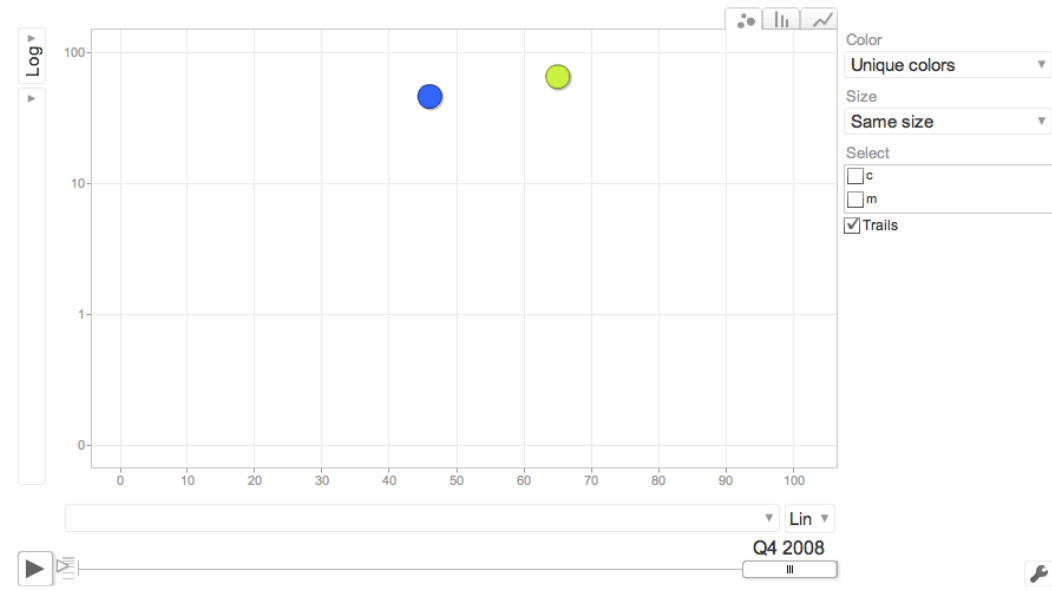
Fig Latin script for finding consistency in Active participants:

```
A = LOAD '2ubuntu-devel-discuss.mbox';
B1 = FILTER A BY $0 MATCHES '.*macoafi@gmail.com.*';
B2 = FILTER A BY $0 MATCHES '.*ciancia@di.unipi.it.*';
... // Similarly we can draw out matches of other emails.
C1_06 = FILTER B1 BY $0 MATCHES '.*2006.*';
C2_06 = FILTER B2 BY $0 MATCHES '.*2006.*';
C1_07 = FILTER B1 BY $0 MATCHES '.*2007.*';
C2_07 = FILTER B2 BY $0 MATCHES '.*2007.*';
... // Similarly for 2008, 2009, 2010, 2011.
D1 = FILTER C3_08 BY $0 MATCHES '.*Jan.*';
D2 = FILTER C3_08 BY $0 MATCHES '.*Feb.*';
```

... // Similarly we can filter the rest of the months.

The visualizations based on the results of these scripts can be viewed below:





Figures: Screen shots of the Motion Chart with flow of Activity of two users.

Check here: <http://www.public.asu.edu/~lmotamar/files/motionChart.html>

This way one can visualize the flow of activity and try to observe any anomalies, if exist any. Right now this is done for just 2 users. When this is done to large number of users, we would find some anomalies.

5.5 Question-3: What are the frequently used keywords (Buzzwords) in the emails?

This is to examine the content of the actual message of the email, from the log file. Here I processed the whole log file and tokenized it into words and calculated the occurrences of each keyword. From these tokenized and measured keywords I created a Cloud of keywords.

Following is the script for tokenizing the words and then calculating their frequencies:

Fig Latin script for finding Buzzwords from sub projects:

A = LOAD '1ubuntu-devel.mbox';

C = FOREACH A GENERATE FLATTEN (TOKENIZE (\$0));

D = GROUP C by \$0;

E = FOREACH D GENERATE group, COUNT(C);

F = ORDER E BY \$1;

Below is table of top most 25 Buzzwords from 2nd sub project.

Keywords / Buzzwords	Frequencies of these buzzwords (number of occurrences).
Ubuntu	27588
Users	14404
Development	12786
Bug	5903
User	5192
Problem	3316
Discussion	3012
Bugs	2765
Ubuntu	3045
Thunderbird	2129
Kernel	1917
Windows	1910
Firefox	1089
Important	1053
Fixed	1042
Configuration	951
Community	945
Discussion	942
Believe	925
Karmic	902
Ubuntu.	843
Unknown	842
Developer	830

Table: Top 25 Buzzwords from second sub project.

Note: These buzzwords are manually selected from top 700 tokenized words.

Note: From the above list, we can observe that words like users and user are considered differently here. And even the words like “Ubuntu”, “ubuntu” and “Ubuntu.” are considered different by the tokenization function.

- ❖ Here not just on the frequencies but we can also measure the tf-idf (Term frequency – Inverted document frequency) of these keywords and determine more accurate measures in determining these buzzwords. This is left as a future update. Here we can consider each sub-project log as a document. And measure the corresponding terms from each of the sub project.

A tree map visualization of the keywords listed before are shown below:

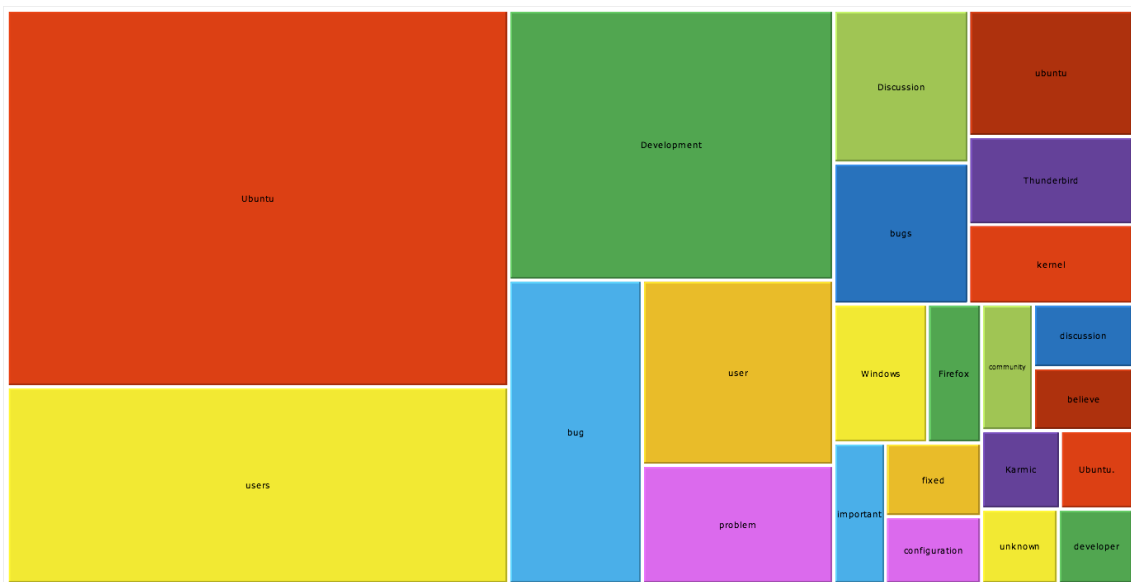


Figure: Top 25 buzzwords from second sub project.

Note: This above tree map is formed with keyword frequencies.

The top 15 buzzwords from 1st sub project are shown in below graph.

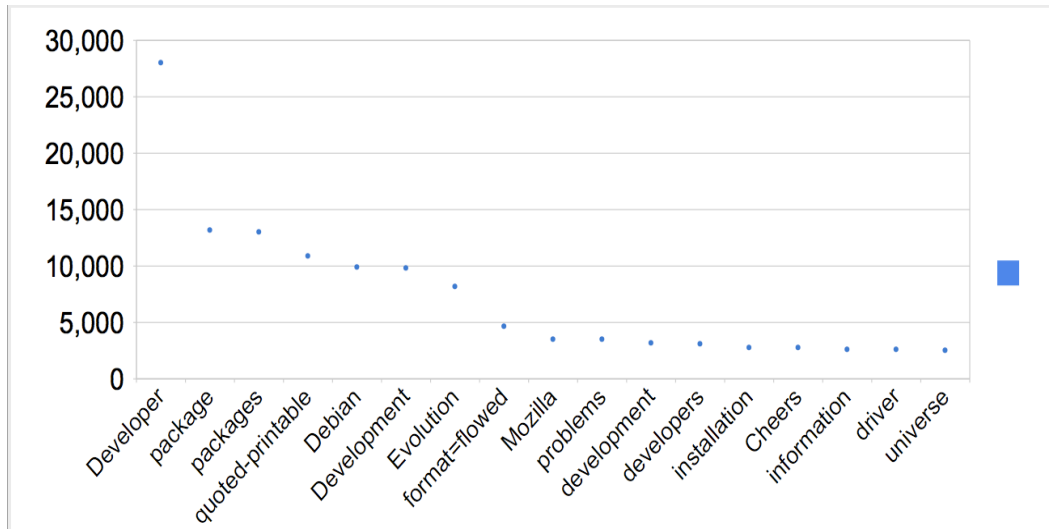


Figure: Graph with 15 buzzwords from “ubunutu-devel” sub project.

Note: These words are manually selected from top 500 tokenized words.

5.6 Question-4: What is the Mood of the community?

Finding the Mood of the community from its email messages, by text classification is what this section talks about. In the earlier question I have found the buzzwords. Similarly I attempted to find the frequencies of few selected moods. Around 132 Moods which were discussed in paper[12] are observed and their frequencies are calculated. These 132 Moods are classified into 5 levels. Each level is assigned an importance measure and based on the computation from their frequencies and measurements, we can decide on the Entire Mood of the community as a whole. This would show the Happiness, Sadness or angriness, etc. of the community.

Pig Latin script for computing Mood Analysis:

A = LOAD '16ubuntu-studio-devel.mbox';

B = FOREACH A GENERATE FLATTEN(TOKENIZE(\$0));

C = GROUP B by \$0;

F = Foreach C GENERATE group, COUNT(B);

happy = FILTER F by \$0 MATCHES 'happy';

angry = FILTER F by \$0 MATCHES 'angry';

awake = FILTER F by \$0 MATCHES 'awake';

...

DUMP angry;

DUMP awake;

DUMP confused;

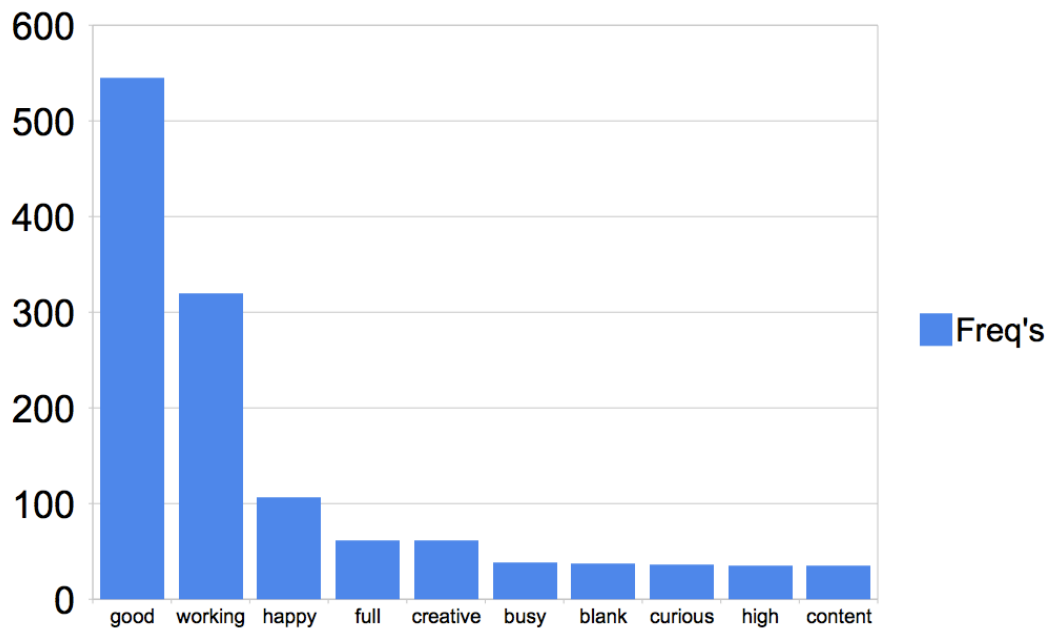


Figure: Top 10 moods of sub-project "ubuntu-studio-devel".

angry	happy	sad	
aggravated	amused	bored	
annoyed	cheerful	crappy	
bitchy	chipper	crushed	
cranky	ecstatic	depressed	
cynical	excited	disappointed	
enraged		discontent	
frustrated		high	
grumpy		horny	envious
infuriated	grateful	good	gloomy
irate	impressed		pessimistic
irritated	jubilant		jealous
moody	loved		lonely
pissed	optimistic		melancholy
stressed		hopeful	morose
			numb
rushed	pleased		rejected
awake	refreshed		sympathetic
confused		rejuvenated	uncomfortable
curious	relaxed		cold
determined	calm		dirty
predatory	mellow		drunk
devious	peaceful		exhausted
energetic	recumbent	drained	
bouncy	satisfied	tired	
hyper		content	groggy
enthralled	complacent		sleepy
indescribable	indifferent		guilty
nerdy		full	hot
dorky		relieved	hungry
geeky	silly		restless
okay		crazy	sick
blah		ditzy	nauseated
lazy		flirty	sore
examine		giddy	thirsty
apathetic		giggly	worried
blank		mischievous	working
lethargic		naughty	accomplished
listless		quixotic	artistic
sacred		weird	busy
anxious	surprised		creative
distressed		shocked	productive
embarrassed	thankful		thoughtful
intimidated	touched		contemplative
nervous			nostalgic
			pensive
	Mood level 1.		Mood level 4.
	Mood level 2.		Mood level 5.
	Mood level 3.		

Figure: 132 Moods considered for the classification.

Note: As of now, for this research work, the negation of these keywords like: ‘Not Happy’, i.e. occurrence of NOT keyword before one of these Mood words is not taken into consideration. For more precise measurement results, we can extend this research by feeding the system not to consider the occurrences of such negations.

Out of these I have classified the 5 level moods and assigned a measure for each level of the Mood.

Level 1 is very frequently used.

Level 2 is less frequently used than level 1 words.

Level 3 are not too frequently and not the words that are rarely used. So they are given high measure.

Level 4 again is quite opposite to level 2. So given similar importance of (*20).

Level 5 are too rarely used words. So given (*10) measure.

		Mood level 3 (*30)		
	Mood level 2 (*20)		Mood level 4(*20)	
Mood level 1 (*15)				Mood level 5(*15)

Figure: Measures considered of each Mood level.

Based on above classification and measurement:

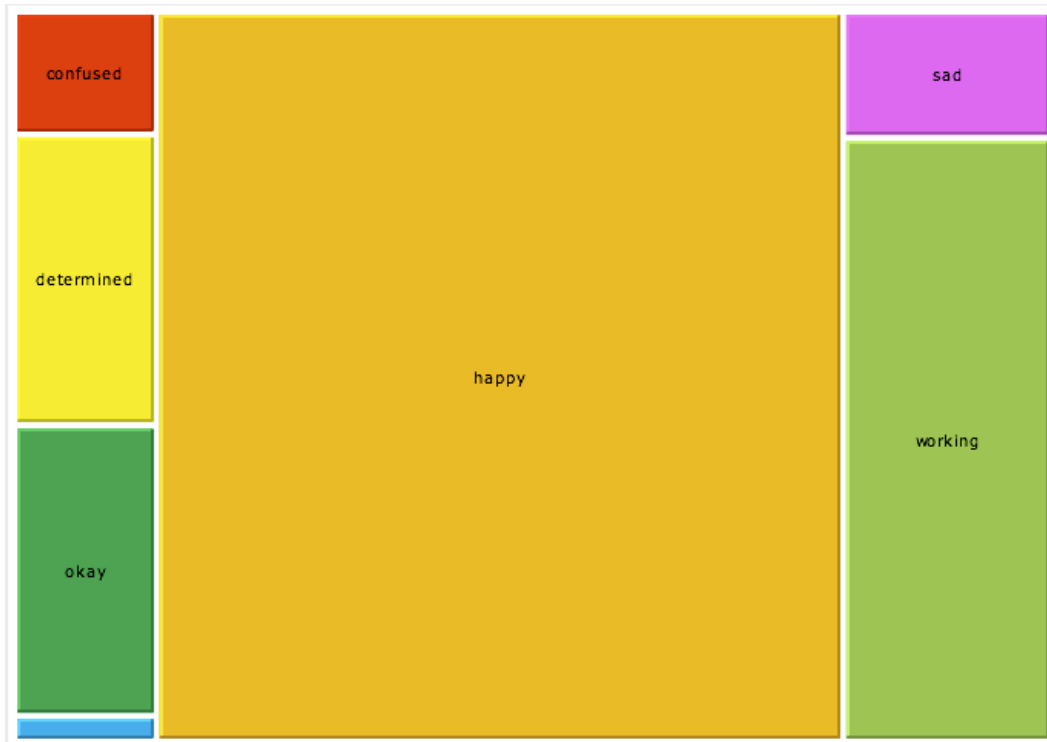


Figure: Top level-1 Moods, with high measurements.

As shown here we can decide the Mood of the entire community from their text email content. This can be inferred that the community happiness shows the satisfaction of the community, interest to further involve in the community.

Both questions 5.7 and 5.8 are left for future agenda.

5.7 Question-5: What are the countries that the participants are from?

This is an attempt to see, the scale of the global community that is using this product.

5.8 Question-6: Try to indentify important communications?

This is an attempt to measure the importance of an email by its thread length (number of replies with same subject name), and then create clusters grouping similar ones together. This way I can visualize the section of important email, from corpse.

CHAPTER 6

CONCLUSION AND FUTURE WORK

The results of five research questions, which I have raised in the Introduction, provide insights into the communications of participants of a mailing list. This approach of measuring features from different perspectives would give you significant results out of Log data.

These results when obtained for all sub projects will give us a chance to correlate the results of each sub project and find any anomalies and then we can analyze the same. As mentioned in 5.6 and 5.7 those are the features that we can measure for more precise insights and are left for future work.

Suitable Areas that can adopt these metrics:

An open source manager can apply these approaches to keep a watch on the performance of open source contributors and also to augment human reporting.

Identifying the active contributors, one can reward based on the findings.

Presenting these insights to the community would create more interest and trust.

These measures and approach with slight modifications can be applied to proprietary software industry as well.

In my opinion, the only limitation as of now is that Pig Latin is relatively new scripting language and is in its 0.9 version. But having said that, Industry has adopted it in a huge level and rate at which it is being adopted and developing it further is also alarming. So I see more scope of scripting precise metrics in the future, which helps us draw more significant and useful insights.

REFERENCES

- 1) Dindin Wahyudin, A Min Tjoa, "Event-Based Monitoring of Open Source Software Projects," pp.1108-1115, The Second International Conference on Availability, Reliability and Security (ARES'07), 2007
- 2) Gloor, P. Laubacher, R. Dynes, S. Zhao, Y. "Visualization of Communication Patterns in Collaborative Innovation Networks: Analysis of some W3C working groups". ACM CKIM International Conference on Information and Knowledge Management, Nov 3-8, 2003.
- 3) Edwards, H. Keith Edwards, Puckett, Robert R., and Jolly, Art. Analyzing Communication Patterns in Software Engineering Projects. Software Engineering Research and Practice 2006: 310-315
- 4) Dindin Wahyudin, Khabib Mustofa, Alexander Schatten, Stefan Biffli, A. Min Tjoa, (2007) "Monitoring the "health" status of open source web-engineering projects", International Journal of Web Information Systems, Vol. 3, pp.116 – 139
- 5) Crowston, K., Annabi, H. & Howison, J. (2003), Defining open source software project success, in 'Proc. of International Conference on Information Systems (ICIS 2003)'.
- 6) Mockus, A., Fielding, R. & Herbsleb, J. (2002). Two case studies of open source software development: Apache and mozilla, ACM Transactions on Software Engineering and Methodology 11 (3), 1 – 38.

7) Kidane, Y. Gloor, P. Correlating Temporal Communication Patterns of the Eclipse Open Source Community with Performance and Creativity, North American Association for Computational Social and Organizational Science, June 26 - 28, 2005.

8) Andresen, E., Bergman, A., Hallen, L. 2006. The role of email communication in strategic networks: patterns observed over time. 22nd IMP Conference. Accessed: 27 November 2008.

9) K. Yelupula, Srinu Ramaswamy, Social network analysis for email classification, Proceedings of the 46th Annual Southeast Regional Conference on XX, March 28-29, 2008.

10) Apache Pig philosophy. <http://pig.apache.org/philosophy.html>

11) C. Olston, B. Reed, U. Srivastava, R. Kumar and A. Tomkins. Pig Latin: A Not-So-Foreign Language for Data Processing. SIGMOD 2008

12) Fazel Keshtkar and Diana Inkpen. "Using Sentiment Orientation Features for Mood Classification in Blog Corpus", IEEE International Conference on Natural Language Processing and Knowledge Eng.(IEEE NLP-KE'2009), Sep. 24-27, 2009.

