Anti-Spam Grid: A Dynamically Organized Spam Filtering Infrastructure

PENG LIU[1,2], GUANGLIANG CHEN[2], LIANG YE[3], WEIMING ZHONG[4]

1 Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

2 Grid Research Center, PLA University of Science and Technology, Nanjing 21007, China

3 Computer Science Department, SuZhou Vocational University, SuZhou 215011, China

4 Network Center, NanChang University, NanChang 330029, China

Abstract: - The spam problem is getting worse all the time. In the paper, we propose Anti-Spam Grid, which can collaboratively filter spam messages by forming a virtual organization. We discuss the design of fuzzy CopyRank and distributed Bayesian algorithm, and describe the architecture of Anti-Spam Grid. A detailed analysis shows that the system is reliable, efficient and scalable, and an experiment shows that the CopyRank mechanism is sharp at distinguishing spam and non-spam.

Key-Words: - spam filtering, Anti-Spam Grid, CopyRank, Bayesian

1. Introduction

Spam is unsolicited bulk e-mails. According a statistics made by Brightmail Probe Network [1], spam accounts for 62% of all Internet e-mail in February 2004, which is 38% more than a year ago. Spam works because of the huge volume of messages sent efficiently at very low cost. A spammer can send out millions of messages in several hours. Responses from a tiny fraction of recipients will make sending it a viable proposition.

Spam is a pricey pest. Nucleus Research found that in 2003 spam will cost the average organization 1.4 percent of employee productivity, or $874 per year per employee [2]. Another survey made by Radicati Group indicates an organization with 10,000 employees spends an estimated $71.91 per mailbox per year because of spam, and the worldwide cost of spam to businesses is expected to be $30 billion this year, and $113 billion by 2007 [3].

The contemporary approaches that fight against spamming include blacklist, good user habit, legislation, cost raising, sender verification and various filtering methods, such as rule-based algorithm [4], Baysian-based algorithm [5], and P2P-based algorithm [6], etc.

This paper will introduce a new anti-spam infrastructure based on the grid technology [7]. By easily forming virtual organizations [8] from distributed servers, we can globally make statistics on e-mails to know more information to better filter spam. The following context will show it is more reliable, efficient and scalable than P2P-based approaches.

The rest of this paper is organized as follows. Section 2 presents the basis of our method, including two presuppositions and the corresponding approaches. Section 3 focuses on architecture of Anti-Spam Grid, and its scalability issue is analyzed in Section 4. A detailed introduction of the other anti-spam approaches is given in Section 5. The final section is the conclusion and future work.

2. Presuppositions and Approaches

Anti-Spam Grid is based on two presuppositions. The No.1 presupposition is that *an e-mail is called a spam only if it is sent to too many recipients*. Thus we assign a CopyRank to each distinct e-mail, which is the number of copies people receive the e-mail. We can now tell whether a new e-mail is a spam or not based on its CopyRank value. CopyRank has the same spirit as PageRank [12] used by Google and ACI (Autonomous Citation Indexing) [13] used by CiteSeer.

The No.1 presupposition can be corroborated by the following facts: Spam has low response rates (on the order of 15 per million), so spammers make up for it with high volumes [11]. Each spammer sends out between 10 to 50 million of spam every day, to address lists scraped from all over the net, obtained from other spammers, copied from spam CDROMs, etc. Over 90% of all the spam received in North America and Europe originates from only about 200 senders [10].

In fact, Brightmail [14] has a similar idea on this, which maintains a network of fake e-mail addresses. Any e-mail sent to these addresses must be spam and can be filtered out when users receive the same e-mail. The system will calculate “signature” for each e-mail and it would be unlikely that a different e-mail would have exactly the same signature. Thus the signature represents the e-mail and can be used to compare whether two e-mails are the same. Unfortunately, spammers can attack the signature-based filter method by adding random stuff to each copy of a spam to give it a distinct signature. As a result, it catches only 50-70% of spam [11].

To avoid this problem, we propose fuzzy CopyRanks instead of accurate ones. E-mails will first be purified by detecting and getting rid of most of the machine-generated “noises”. Then we use an algorithm to generate checksums that will let similar e-mails have close values. The overall CopyRank of an e-mail will be calculated out from a set of neighboring CopyRanks. Though the length of the checksum is large enough to minimize the possibility of different e-mails have the same checksum, it is still possible when the number of e-mails increase. So, we should aging the checksums. By these means, there's a fairly high chance of overcoming the tricks used by spammers to add random stuff to e-mails.

The No.2 presupposition is that *the information gathered by many computers will be more accurate and complete than that from only one computer*. Thus we will apply a distributed Bayesian filtering algorithm, which will carry out Bayesian studying process among hundreds of thousands of client computers and then collect and spread the up-to-the-minute information to all active clients. Also, servers can make a great contribution to this process by setting up numerous fake e-mail accounts to attract spam and then screen it.

Moreover, we have proposed an improved Bayesian algorithm, in which not only words or phrases are regarded as tokens, but also some special characteristics of e-mail are introduced into distributed Bayesian model. For example, since about 95% of spam contain links to web pages [11] and the links cannot be disguised, then this information will have a heavy weight in the Bayesian model if it appears in a e-mail that have a high CopyRank value. Other methods used by rule-based filters can also be integrated into our Bayesian algorithm, such as whether the user has ever received legitimate e-mails from the sender before, whether the pictures or attachments in e-mails are the same, etc. Incidentally, some viruses that spread by e-mails often has attachments or links of the same type, so they are likely to be fenced out by the filter similarly.

The fuzzy CopyRank and distributed Bayesian algorithm will have many advantages. First, new users needn’t to train the filter before use, since the statistics information got from the other computers and servers is enough for a good start. Second, the network will be very sensitive to any new-style spam that goes beyond the statistics of Bayesian model, since their sharply increasing fuzzy CopyRank value will alert all clients. Third, the whole system will be evolving all the time, studying on every new e-mail. Finally, the scheme can prevent the filters to be “false positives”, since it will never regard an e-mail with low CopyRank value as a spam, even though it is full of *bad* words such as “Guaranteed” and “Free”, etc.

3. Anti-Spam Grid Architecture

We believe that the grid technology with OGSA (Open Grid Services Architecture) [9] is a good choice for our idea, based on the following considerations: (1) Spam is delivered globally, so we need a global infrastructure to gather information on spam. (2) As central control system may result in bottlenecks, a collaboration of distributed services will efficiently serve local users. (3) It is a most dynamic environment that all the servers, clients, and e-mails keep changing all the time, so we need to form a virtual organization which is adaptive to changes.

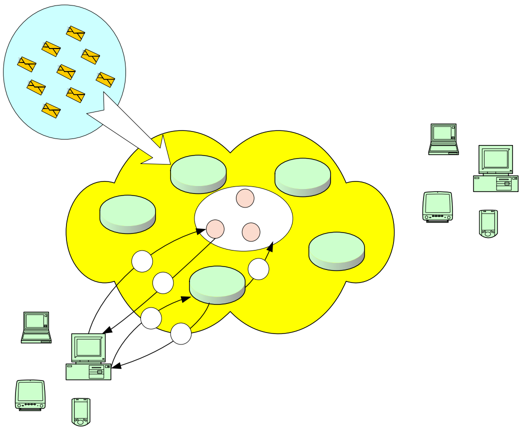

Figure 1 Architecture of Anti-Spam Grid

The architecture of Anti-Spam Grid is shown in Figure 1. At the time of starting up, anti-spam servers will publish their services to one of the dispatchers, which act as information center and server assigner when a client first accesses the grid. The dispatchers will exchange their information at intervals to be almost coherent to each other. Some servers may setup many fake e-mail accounts to capture pure spam.

A user can join the grid by downloading a special plug-in for his e-mail client program, e.g. Outlook Express. During its setup process, the plug-in will connect one of the dispatchers to find the nearest server in the grid. Once the server is assigned, the user will use it forever, unless it fails to provide a QoS assured service for the user.

Every time the user receives an e-mail, the plug-in will be activated. After doing calculation for the e-mail, the plug-in will send the checksum to the selected server, which will feed back a CopyRank value for the checksum and some statistics information gathered from fake e-mail accounts and other clients. The filter then can perform an improved Bayesian algorithm based on the CopyRank value to decide whether to flag an e-mail as a spam or not.

As each server only faces part of the clients, it is necessary to gather information from all the servers to maintain a complete checksum-CopyRank table that reflect the global situation. This can be done by dynamically selecting a server as a temporary center, which will gather and summarize changed information from each server and then spread the latest global information to all the servers. As each server will have a copy of the global information, malfunction of any server, including the temporary center, will only affect an infinitesimal part of the global information.

4. Scalability Issue and Analysis

From the above section, we can see the advantage of using grid platform: both the dispatchers and the servers can be dynamically added to the system along with the growth of the volume of users. As the servers will register in the dispatchers, it is easy for the dispatchers to redirect part of the users to any newly added servers. But how can the users’ plug-in know the address of the dispatcher? Fortunately, as the number of dispatchers only grows with the number of users, we can assign a different primary dispatcher address for every *T* copies of download. Also, we can assign several backup addresses in case of any failure of the primary dispatcher. The following context introduces the way to figure out the value of *T*.

Suppose *P* is the maximum processing power of a dispatcher, which means the largest number of users it can serve per unit time. New users arrive at a speed of *A(t)*, and the overall number of users in the system at time *t* is *T(t)*. Any new user will be redirected to a server by the dispatcher and will not come back unless the server fails at the probability of *f*. Since we want to calculate the maximum user capacity of one dispatcher, it is reasonable to suppose the dispatcher is saturate all the time. Thus we have:

$$P = A(t) + f \cdot T(t) \qquad (1)$$

Actually, it is possible that some users of *T(t)* may leave the system. Still, the equation is correct by regarding *A(t)* as the difference between arriving rate and leaving rate.

Though *P*, *A(t)* and *T(t)* are discrete values, they can be regarded as continuous values when the number of users is large enough. We can find a relationship between *T(t)* and *A(t)*:

$$T(t) = \int_0^t A(x)dx \qquad (2)$$

Then (1) can be rewritten as:

$$P = \frac{dT(t)}{dt} + f \cdot T(t) \qquad (3)$$

This is a differential equation which can be solved as:

$$-\frac{1}{f}\ln(P - f \cdot T(t)) = t + C \qquad (4)$$

where C is a constant. By using the boundary condition:

$$T(t)\big|_{t=0} = 0 \qquad (5)$$

we can figure out:

$$C = -\frac{1}{f}\ln(P) \qquad (6)$$

Then we can get the solution of *T(t)* from (4):

$$T(t) = \frac{P \cdot (1 - e^{-ft})}{f} \qquad (7)$$

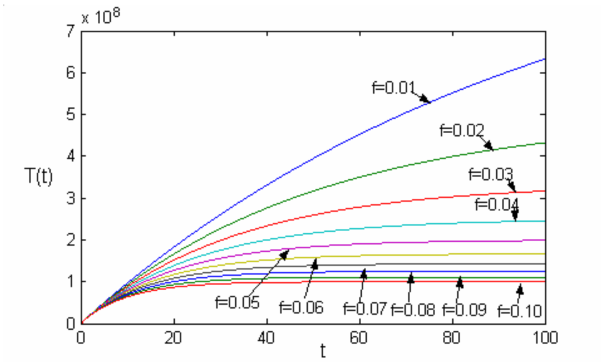

Figure 2 Volume of the system as time varies

Formula (7) is illustrated as Figure 2. The time unit is a day, and *P* is assigned a value of 1×10^7, which is the number of requests that a dispatcher can handle in one day. The value of *f* varies from 0.01 to 0.10, which means 1% to 10% servers may fail in one day. It is obviously a conservative estimation. If *f*=0.03, for example, the dispatcher will be saturate after 100 days running when there are 3×10^8 users. So we can select *T* according to this value.

5. Experiment

Experiment has been conducted in a LAN environment, in which 4 nodes are configured as Mdaemon e-mail server with our anti-spam SMTP filter, 3 nodes are configured as spammer's machines, and 1 node acts as dispatcher. Each node has a Ce 1.7 CPU, 256M memory, and has been installed Windows 2003 Server. In the experiment, filters are installed in the e-mail server sides, instead of the client sides. Nevertheless, the logics are the same.

The experiment lasts 200 minutes with 40,000 sample e-mail files processed, 60 percent of which are set as spam according to Brightmail [1]. The result is shown in Figure 3, in which LikelySpam means e-mails with a CopyRank between 20 and 100, and Spam means e-mails with a CopyRank above 100. From the experiment, we can see that almost 100 percent of the spam are flagged correctly with the CopyRank mechanism.

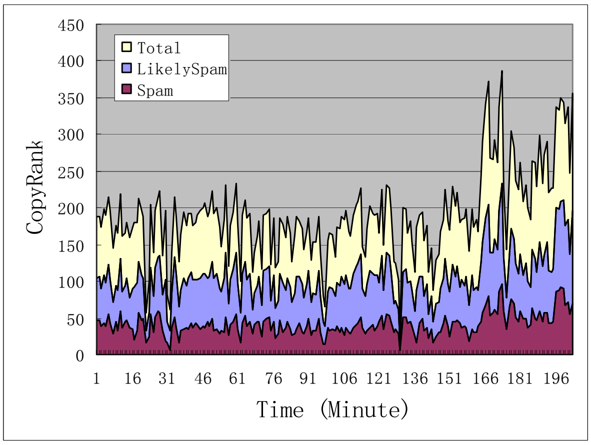

Figure 3 Spam and LikelySpam found out

6. Related Work

Blacklist is a traditional and simple way to filter spam. It is hard to be effective since most spam will forge their source addresses. User's good habit, however, may be helpful. For example, if you never divulging your e-mail address online, you will less chance be known by spammers. Furthermore, Anti-spam legislation could eliminate 80% of spam.

There are also some proposals on raising cost of spamming, such as the FFB (Filters that Fight Back) [15], as well as the Slow Senders method and Penny per Mail method. There are also ways to verify the senders, such as the ePrivacy Group's Trusted E-mail Open Standard [16] and Challenge-Response Filtering

Compared to the above approaches, local filtering policies seem to be better adopted by people. A simple rule-base filtering mechanism, will block messages that match pre-specified criteria, based on *bad* words in the title or body, the sender’s address, and the length of the e-mail, etc. Sophisticated rule-based filters, like Spamassassin [18], assign each message a score based on patterns including specific words and phrases, lots of uppercase and exclamation points, malformed headers, dates in the future or the past, etc. If it is above a specified score, messages are flagged as spam. Nevertheless, as rule-based filters are static targets, spammers can tune their mails to get through them.

On the contrary, a Bayesian spam filter like SpamProbe [17] is an adaptive filer that turns to be smarter and more accurate. It will make statistics from both spam and legitimate e-mails. Then the filter can calculate a probability score to be a spam for a new e-mail. The disadvantage of Bayesian filters is that they need to be trained for hundreds of examples and still, spammers have got tricks to bypass the fence, for example, to send all-graph e-mails which contain no significant tokens.

Most filters are double-edged swords: when they are configured stringently enough to be effective, there is similar possibility that they are getting *false positives*: legitimate e-mail is misclassified as spam. The small chance of accidentally blocking an important message has been an enough reason to keep users from filtering spam at all.

The P2P-based approaches [6] will first use hash functions to generate digests of e-mails, and then lookup the digests from a centralized or decentralized repository. The main drawbacks of P2P system are: the decentralized autonomous nature leads to unpredictable performance, reliability and safety; and the relatively complicate routing and searching methods will wear down its efficiency.

7. Conclusion and Future Work

We present design and evaluation issues of Anti-Spam Grid, an infrastructure dedicated to filter unsolicited bulk e-mails. Based on the grid technology, dispatchers and servers can be dynamically added to the system as user volume grows. At the same time, the system can run properly whenever any dispatcher or server fails. So we can say the system reflects the core idea of the grid: virtual organizations. Analysis and experiment show that it is very effective and scalable.

Our future work includes study on a low cost method to keep the data in distributed servers to be near coherent, and ways to be dead against any future improvement of spamming algorithms. The goal of our research is to deploy Anti-Spam Grid [19] as a long-running service and an infrastructure for the public.

8. Acknowledgements

This work was supported by the China National Science Foundation with Grant No.60403043.

References

[1] Spam Attacks and Spam Categories, http://www.brightmail.com/spamstats.html

[2] Nucleus Research , Spam: The Silent ROI Killer, http://www.nucleusresearch.com/research/d59.pdf

[3] Jon Panker, Spam is a pricey pest, http://searchnetworking.techtarget.com/originalContent/0,289142,sid7_gci902228,00.html

[4] Cohen, W. W. Learning rules that classify e-mail.In: *Proceeding of the AAAI Spring Symposium on Machine Learning in Information Access*, 18-25, 1996

[5] Sahami, M. Learning limited dependence Bayesian classifier. In: *Proceeding of the Second International Conference on knowledge Discovery and Data Mining*, 335-338, 1996

[6] Feng Zhou, Li Zhuang, Ben Y. Zhao, Ling Huang, Anthony D. Joseph and John Kubiatowics, Approximate Object Location and Spam Filtering on Peer-to-peer Systems, In: *Proceeding of ACM/IFIP/USENIX International Middleware Conference,* 2003

[7] Ian Foster, Carl Kesselman, ed. *The Grid 2: Blueprint for a New Computing Infrastructure, 2nd edition, Morgan Kaufmann,* Nov. 2003, ISBN: 1558609334

[8] Foster, I., C. Kesselman, and S. Tuecke, The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International Journal of High Performance Computing Applications*, 2001. 15(3): p. 200-222.

[9] Foster, I., Kesselman, C., Nick, J. and Tuecke, S. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Globus Project, www.globus.org/research/papers/ogsa.pdf, 2002

[10] The Spamhaus Project, http://www.spamhaus.org

[11] Different Methods of Stopping Spam, http://www.secinf.net/anti_spam/Stopping_Spam.html

[12] S. Brin and L. Page, The anatomy of a large-scale hypertextual Web search engine, *Computer Networks and ISDN Systems*, 30,107--117, 1998.

[13] Steve Lawrence, C. Lee Giles, Kurt Bollacker, Digital Libraries and Autonomous Citation Indexing, *IEEE Computer*, Volume 32, Number 6, pp. 67-71, 1999.

[14] Brightmail Probe Network, http://www.brightmail.com

[15] Filters that Fight Back, http://www.paulgraham.com/ffb.html

[16] ePrivacy Group, http://www.eprivacygroup.com/

[17] Spamprobe – A Fast Bayesian Spam Filter, http://spamprobe.sourceforge.net/

[18] Spamassassin, http://au2.spamassassin.org/doc.html

[19] Anti-Spam Grid, http://www.antispamgrid.net