An Algorithm for Web Firewall Packet Scheduling Based on Delay Prediction

Shichang XUAN, Kaige AN, Wu YANG*

Information Security Research Center, Harbin Engineering University, Harbin, Heilongjiang, 150001, China

***Abstract* — As a kind of firewall working at the application layer, web firewall has been widely adopted for its complete functionality. Due to working principle of web firewall the tedious process with packets reduces its throughout. Web firewall's throughput is much lower than packet filtering firewall. An algorithm based on access history is presented in this paper, which is used to estimate delay limits of high frequency URL sessions. Its feasibility is verified through experiment. To take advantage of the upper statistical results, the output queue uses application layer session feedback to schedule packets to improve session delay. Finally, the paper compares the average delay of the algorithm with delay estimation and no delay guarantee. The results show that the algorithm with URL history approach can reduce the occurrence of long delay in less traffic, and it can effectively reduce high frequency URL session delays in large flows. Finally, the correctness of the method is verified by experiments.**

Keywords-web firewall; bloom delay prediction; delay distribution

I. INTRODUCTION

In recent years, with the rapid development of Internet, web applications have become the main carrier business system by its unique efficiency, usability and timeliness which make a lot of institutions transfer their business to web application layer. E-commerce, e-government, online banking and social networking make web applications become an indispensable part of people's life. They are the main channel of access to information. However, with the rapid development of web applications, the security situation is not optimistic, security risks of web application layer is more and more serious. According to the system network security association SANS statistics, 60% of cyber attacks occurred in the web application layer. Web firewall is the best solution to protect web application [1]. However, due to web firewall working at application layer, and faces with larger web flows, its complex packet processing mechanism causes large delay of session. This will make users surfing on the Internet feel the slow network. Serious when, it also can cause network congestion. So how to avoid the occurrence of the above situation, it has become a new hot topic.

Web session delay is associated with a TCP connection. Web applications based on TCP protocol, use the reliable data transmission channels TCP provides. Strict web session delay is defined as the data transmission delay between the beginnings to the ends. Because web data transmission delay is much larger than TCP connection establishment and disconnection, we can make the delay between connection establishment and disconnection as web session delay. In order to improve web session delay, web firewall modules need to cooperate with each other. As an important modules of web firewall, packet scheduling mechanism must be researched.

The article analyzes the web session delay and then puts forward the packet scheduling algorithm based on time delay estimation. The paper mainly includes web firewall packet scheduling algorithm based on time delay prediction of analysis, the principle of algorithm and the analysis of result of the experiment.

The remainder of this paper is organized as follows. "Related Works" explains the research directions of associated work and elucidates the innovation of the present work. "Algorithm Analysis", the principle of the algorithm is proposed. "Experimental Evaluation", the experiment is carried out to verify the algorithm. "Conclusions" summarizes and analysis the algorithm.

II. RELATED WORK

The part of network packets from the source to the end-to-end delay can be optimized including queuing delay and processing delay. In queuing delay, at present, there are two kinds of common scheduling mechanisms of buffer management: Round-Robin (RR) algorithms and Earliest Deadline First (EDF) algorithm. Round-Robin algorithms are the simplest scheduling algorithms. The scheduler retrieves a task from a task queue every time through polling the task queue, which can guarantee the equality of the equity between the queues [2]. When applied to store-and-forward model, we can regard task as packet, the task queue can be regarded as the output buffer, and the scheduler can be regarded as packet forwarding thread. This model can guarantee the fairness between queues. Earliest Deadline First is a scheduling algorithm according to max queuing delay packet requests. It adopts greedy algorithm and makes packet's priority in buffer change with packet deadline and packet count. The closer of packet deadline, the higher its priority, which shows the Shortest-Job-First [3]. In this study, the firewall processing delay is included in the processing delay.

In this paper, we use the knowledge of queuing theory, and many scholars have put forward the concept of queuing theory [4-7]. In [8-18], many of the researchers focused on

the development of firewall technology and performance optimization.

Based on the above research, analyzes the packet processing in web firewall and extracts the session delay by taking use of the session time interval between connection and disconnection. An effective adaptive packet scheduling algorithm based on session delay prediction is put forward by research on packet queue theory.

III. ALGORITHM ANALYSIS

This chapter analyzes the packet processing in web firewall and extracts the session delay by taking use of the session time interval between connection and disconnection. An effective adaptive packet scheduling algorithm based on session delay prediction is put forward by research on packet queue theory. The algorithm combines the extracted URL of the web firewall in HTTP layer [19] and weights of send buffer scheduling to reduce the delay influence of web firewall's complex processing with high frequency URL.

A. Delay Prediction Principle

According to packet transmission process in the network, packet delay between source to end is as follows:

(1) Transmission Delay: Time consumed during the source send data in the local link. It is proportional to the packet size and inversely proportional to the local link rate.

Suppose there is a packet, its size is L, and the link rate is R, then the transmission of the packet will be $\Delta t_{trans} = \frac{L}{R}$.

(2) Propagation Delay: Time consumed from sender to receiver in the link, which is associated with transmission medium and length of link, has nothing to do with packet size and packet processing delay of device.

(3) Queuing Delay: During data transmission, the link bandwidth is limited to the sender. When the transmission speed is greater than the link bandwidth or confliction occurs, the source can't send data in the link timely, which can lead packets accumulation in the queue of sender and queuing delay is coming. For the store-and-forward equipment, the packet should wait for upper processing in the input buffer during the processing of packet transmission. The packets will accumulate in input buffer when upper processing speed is less than the traffic arrival rate. Packets will get service until front packets are processed. The time packet waits in the input buffer is queuing delay and queuing delay also exists in output buffer. So, queuing delay is composed of input and output queuing delay for store-and forward queuing model [20].

(4) Processing Delay: This part is mainly caused by the router, include search path and the packet access process. When there is a firewall in transmission path, it is also affected by the firewall.

For each web session, the whole session delay is composed of several network packet delay. They are closed related with each other.

According to the classic queuing theory model, the packet arrival rate is in conformity with the Poisson distribution. The average service time of packet is $\bar{\Delta t}$, so packet queuing delay in the queue is approximate Poisson distribution. Considering the characteristics, web session delay consists of two parts, the fixed time delay and approximate Poisson distribute delay.

According to the nature of Poisson distribution, superposition of two independent Poisson distributions is a new Poisson distribute. Here is the proof:

Suppose distribution of packet p_1 and packet p_2 are $\{N_1(t), t \ge 0\}$ and $\{N_2(t), t \ge 0\}$, and their strength are λ_1 and λ_2. Then their superposition will be follow:

$$
\begin{aligned}
&P_1\{N(t_2)-N(t_1)=k\} \\
&=P\{[N_1(t_2)+N_2(t_2)]-[N_1(t_1)+N_2(t_1)]=k\} \\
&=P\{[N_1(t_2)-N_1(t_1)]+[N_2(t_2)-N_2(t_1)]=k\} \\
&=\sum_{i=0}^{k} P\{N_1(t_2)-N_1(t_1)=i, N_2(t_2)-N_2(t_1)=k-i\} \\
&=\sum_{i=0}^{k} P\{N_1(t_2)-N_1(t_1)=i\}\cdot\{N_2(t_2)-N_2(t_1)=k-i\} \\
&=\sum_{i=0}^{k} \frac{[\lambda_1(t_2-t_1)]^i}{i!}\exp[-\lambda_1(t_2-t_1)] \\
&\quad\cdot\frac{[\lambda_2(t_2-t_1)]^{k-i}}{(k-i)!}\exp[-\lambda_2(t_2-t_1)] \\
&=\sum_{i=0}^{k} \frac{\lambda_1^{\,i}\lambda_2^{\,k-i}}{i!}\cdot(t_2-t_1)^k\cdot\exp[-(\lambda_1+\lambda_2)(t_2-t_1)] \\
&=\frac{1}{k!}\sum_{i=0}^{k} C_k^i\lambda_1^{\,i}\lambda_2^{\,k-i}(t_2-t_1)^k\exp[-(\lambda_1+\lambda_2)(t_2-t_1)] \\
&=\frac{[(\lambda_1-\lambda_2)(t_2-t_1)]^k}{k!}\exp[-(\lambda_1-\lambda_2)(t_2-t_1)]
\end{aligned}
\tag{1}
$$

According to the composition of web session delay, its distribution is the session superposition of several packet delay distribution. So, web session delay distribution should be similar to the Poisson distribution. Figure 1 is delay distribution histogram of 1000 times on the same page access.

Figure 1. Web session delay distribution

Table 1 shows the delay probabilities of different scopes. According to the data of table 2, web session delay and Poisson distribution model has higher fitting degree. A large number of web session delay distribution can be as a Poisson distribution.

TABLE 1. WEB SESSION DELAY CUMULATIVE PROBABILITY

delay limit	56	59	62
probability	0.024	0.054	0.027
delay limit	65	68	71
probability	0.566	0.768	0.874

Table 2 standard Poisson distribution can be described as follow:

$$P(X \le x) = \sum_{k=0}^{x} \frac{\lambda^k e^{-\lambda}}{k!} \tag{2}$$

TABLE 2. STANDARD POISSON DISTRIBUTION

X \ λ	1.0	2.0	3.0	4.0	5.0
0	0.6689	0.2461	0.0906	0.0333	0.0123
1	1.0	0.5168	0.2400	0.1066	0.0460
2	1.0	0.7876	0.4641	0.2532	0.1302
3	1.0	0.9681	0.6882	0.4486	0.2707
4	1.0	1.0	0.8563	0.6441	0.4462
5	1.0	1.0	0.9571	0.8004	0.6218
6	1.0	1.0	1.0	0.9047	0.7681
7	1.0	1.0	1.0	0.9940	0.8726
8	1.0	1.0	1.0	1.0	0.9379
9	1.0	1.0	1.0	1.0	0.9742
10	1.0	1.0	1.0	1.0	0.9923
11	1.0	1.0	1.0	1.0	1.0

Based on the above facts, if given a certain amount of web session distribution of a random samples. We can conclude the maximum time-delay in the near future according to a set of expecting probability value p.

B. Time Delay Prediction Method

According to the principle of analysis, if you can get recent web session delay, max session delay can be effectively predicted. And measures can be took to process the session's packet preferentially, ensures session time delay.

To avoid the interference of other factors on the session time delay, the algorithm takes [client IP, URL] as the key value to distinguish web sessions.

First: Counting amount of [client IP, URL] in web firewall and the TCP quad information associated with current session. Time delay of current session can be calculated according to start and end of the session.

Second: Searching the URL session cache to find whether there is the same item as current session. If found, corresponding session delay information in the cache should be updated. The current cache entry access time recently should be updated, its average delay should be calculated and interval of time delay should be calculated according to min delay and max delay. Otherwise, the oldest URL item in cache should be replaced by current session's URL.

Third: Unconcern session cache can be constructed with the quad of URL item in URL cache, which will notify the bottom don't send its packets to the upper. This can reduce the network congestion phenomenon caused by the upper bottleneck. At the same time, weight of send buffer related with URL session should increase to reduce queuing delay of the unconcern session.

Description of how to update delay buffer with URL session information is as follow:

Algorithm 1 URL session delay update

Input: recent session information
(url, TCP quad, session)
Output: URL cache updated
Begin
The new finished URL session $U_{new}(url, ip)$
Searching in URL cache according to [url, ip] two-tuples of new URL session
If found
If URL cache is full
Delete the oldest URL item and insert information of U_{new}
Else
Insert information of U_{new}
Update max delay $d_{\max}$, min delay $d_{\min}$, average delay d_{avg} and recent updating time t_{update} of the URL item.
Else
Find the min t_{update} and replace it by t_{update} of U_{new}
At the same time, update its max delay $d_{\max}$, min delay $d_{\min}$, average d_{avg} and t_{update} of new URL item
End

According to the nature of Poisson distribution, items can be divided into N sections with max delay $d_{\max}$, min delay $d_{\min}$ and average d_{avg}. The i-th section corresponding to Poisson distribution's parameter 'k'. To ensure a relatively low error and that time delay of new session is in a range related with early statistical delay interval, delay bounds probability p is took here. The large p, the large delay bound. For a processing HTTP session, if its session delay is larger than historical delay, the flow should be marked as a forwarding flow. When subsequent packets of the flow are placed in the output queue, increasing the weight of corresponding output queue can ensure the session be forward preferentially, so as to ensure the time delay of the session. Algorithm is described as follows:

Algorithm 2 updating weights of output queues

input: packet belongs to useless session

output: priority q_i of queue, length l_i of queue

Begin

If the packet belongs to useless session

Increase weight of output queue, let $q_i = q_i + 1$

Increase length of output queue l_i,

let $l_i = l_i + 1$

Else

Increase length of output queue l_i,

let $l_i = l_i + 1$

End

Algorithm 1 and 2 make up the main part of URL time delay prediction and the part can described by algorithm 3.

Algorithm 3 URL item updating and weights of queue management

```
Input: packet contains HTTP layer content
output: weight q_i of output queue
Begin
if(find_session(packet)){
   flow = find_flow(packet);
   if(session's time larger than max delay){
      flow.pri = true;
   }
   else{
      if(session_end(packet)){
         update-url-cache(flow);
      }
   }
}
else{
   if(session_end(packet)){
      flow = find_flow(packet);
      node = build_new_node(flow);
      if(!cache_full()){
         insert_cache(node);
      }
      else{
         replace(oldest_node, node);
      }
   }
}
if(the packet's flow.pri is true){
   let q_i ++;
}
End
```

In algorithm 3, if session delay is greater than the maximum limit, the flow of the session will be marked as forwarding flow. Packets of forwarding flow won't be delivered to application layer, but will be put in output buffer directly.

During update-url-cache, the minimum, maximum and average of URL item will be updated. The set of historical session delays can be divided by step Δt with its minimum and maximum. Then the recent maximum delay bounds can be found by search table 2 with given probability p.

On condition that there is a session '[192.168.100.42, www.baidu.com/index.html] whose minimum delay is 76 milliseconds, maximum delay is 396 milliseconds and average delay is 112 milliseconds. The set of session delay can be divided into 20 sections with step 16 milliseconds. Suppose that expected probit p=0.98. According to table 2, session delay bounds is 172 milliseconds.

C. Packet Scheduling Based on Delay Prediction

With the session delay bounds calculated in last section, if session's delay is greater than maximum delay of its item, the session should be forward by web firewall. According to the principle of firewall packet scheduling, FIFO queue scheduling is a familiar method. To improve delay of session that is greater than maximum delay bounds, weight of packet send buffer can increase. Then packet in the relevant buffer can be forward preferentially and session delay can be improved.

In this algorithm, every send buffer k has a weight q, they are $q_1, q_2, \ldots, q_k$. The basic composition unit of session is packet rather than byte, so weight represents maximum number be serviced, that is limit of number packets should be sent all at once.

Algorithm 4 packet scheduling based on feedback of session

Input: count of packets l_1, l_2, l_3 …

weight of queues q_1, q_2, q_3 ...

output: amount of remnant packets in output

buffer l_1', l_2', l_3', ..., and weight of output

buffer q_1, q_2, q_3

PSOS($\sum_{i=1}^{k} l_i$, $\sum_{i=1}^{k} q_i$)

```
Begin
  set i = 1;
  while(i < k){
    If l_i > 0{
      If l_i > q_i {
        Send q_i packets in ith queue;
        let l_i' = l_i - q_i;
      }
      Else{
        Send l_i packets in ith queue;
        let l_i' = 0;
      }
      set q_i' = 1;
    }
    let i = i + 1;
  }
End
```

At the start, every queue has weight 1, which presents the queues have the same priority. When the scheduling thread serves queues, it will consume a packet of time. Its initial state is similar to RR algorithms. However, weights of queue is a dynamic change value with session of application layer. When there is a session with higher priority, the weight of relative output buffer should increase. The buffer can get more services of scheduling thread later and queue delay of the session can reduce, session delay can reduce.

IV. THE EXPERIMENT RESULT ANALYSIS

The experiment is performed in local network to simulate principle of web firewall and two models are used about output buffer. Throughput changes and packet loss rate changes are contrasted with the two models. The network environment of experiment are as follows:

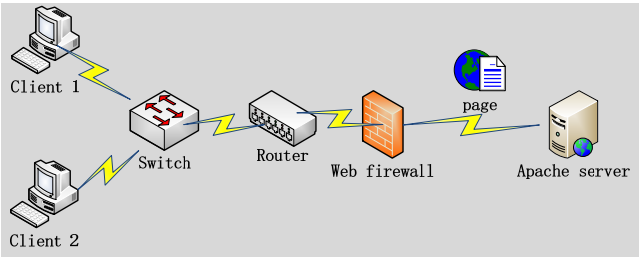

Figure 2. Topology of network

The bandwidth of apache server is 1000Mbps, and it is used to store web document. The clients test the apache server with apache stress test tool. From the above figure, it is obvious that the traffics of client flow through switch, web firewall to apache server. Then the server return documents to client through web firewall and switch.

Parameters of devices in this experiment are as table 3.

TABLE 3. PARAMETER CONFIGURATIONS OF DEVICES

Type of device	Memory size	bandwidth	IP address
Client 1	3.33GB	100Mbps	192.168.100.56
Client 2	3.33GB	100Mbps	192.168.100.57
Firewall	512MB	1000Mps	192.168.100.241
Apache server	16GB	1000Mps	192.168.13.240
switch	None	1000Mps	None

In this experiment, the clients test congestion state of web firewall with a stress test tool for http server, http_load. The software is a tool for web firewall and has character of parallel running. It is used to test the maximum throughput and concurrency of web server. The system of client is Windows and the version of http_load is 2.0. The tool can be used to test concurrent capabilities of server and control the flow. For example:

http_load –parallel 5 –s 60 url.txt

As above, url.txt stores testable url address, that is the path of document. 5 threads will be created in this command and they will run 60 seconds for getting documents in the file uninterruptedly.

The test is carried out in local network and delay distributions about delay prediction algorithm and others are contrasted. At the same time, the algorithm's effect can be reflected by loss packet rate changes.

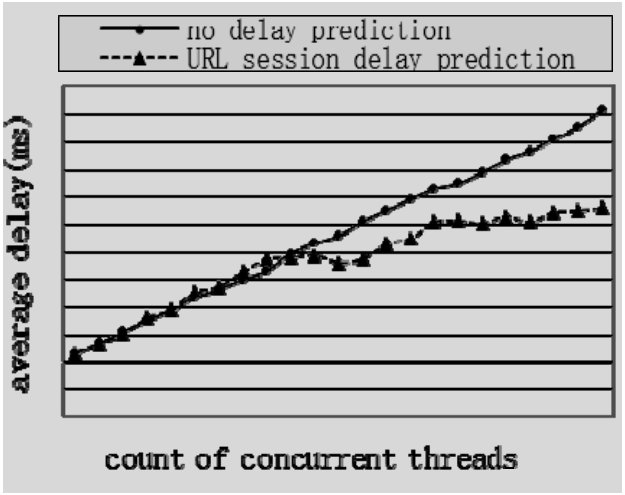

Figure 3. Average delay changes

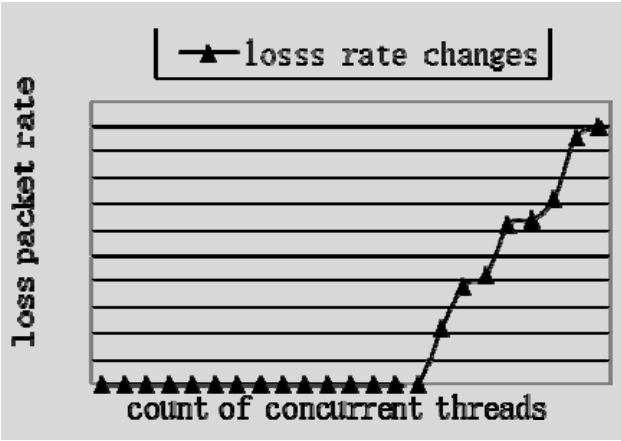

Figure 4. Loss rate changes

Figure 3 shows that no matter URL session delay prediction algorithm or other algorithms is used, their session delay increases with count of threads increases. However, the former increases flatly and the latter steeply. The growth trend of URL delay prediction is much slower than latter when firewall is conjunctive(Figure 4). The figures profs that URL session delay prediction can effectively guarantee the session time delay and reduce firewall's influence on the session time delay especially when load is bigger.

V. CONCLUSION

This study analyzed the process of web session and packet transmission, proposed session delay prediction algorithm based on delay statistics by using the characteristics of packet transmission and the properties of Poisson distribution. By simulation in local network, its correctness is verified. The article improved WRR algorithm when scheduling packets in output queues. The new packet scheduling algorithm updates queue weight by information of application session that is timeout. Using the method, packets without obvious priority and sessions with excessive delay can be forwarded preferentially. Finally, the correctness of the method was verified. The experiment results show that URL statistical method can reduce phenomenon of URL with large delay and session time delay can be guaranteed effectively when load increases.

ACKNOWLEDGMENTS

This work was funded by the Fundamental Research Funds for the Central Universities (HEUCF160605).

The authors would like to thank the editors and anonymous reviewers for their valuable comments and feedback, which greatly improved the quality of this paper.

REFERENCES

[1] SANS Institute, "Top Cyber Security Risks-Executive Summary," Available on http://www.sans.org/top-cyber-security-risks/summary.php, 2009.

[2] C. C. Chude-Olisah, U. A. Chude-Okonkwo, K. A. Bakar and G. Sulong, "Fuzzy-based dynamic distributed queue scheduling for packet switched networks," Journal of Computer Science and Technology, vol.28, pp. 357-365, 2013.

[3] A. Ioannou and M. G. Katevenis, "Pipelined heap (priority queue) management for advanced scheduling in high-speed networks," IEEE/ACM Transactions on Networking, vol.15, pp.450-461, 2007.

[4] L. Kleinrock, "Queueing systems, volume I: theory," 1975.

[5] R. Jain, "The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling," John Wiley & Sons, 1990.

[6] H. Takagi, "Queueing Analysis, Vol. 1: Vacation and Priority Systems," NorthHolland, Amsterdam, 2010.

[7] D. Gross, "Fundamentals of queueing theory," John Wiley & Sons, 2008.

[8] A. Mayer, A. Wool and E. Ziskind, "Fang: A firewall analysis engine," Security and Privacy, 2000 S&P 2000 Proceedings 2000 IEEE Symposium on, 2000.

[9] J. Qian, S. Hinrichs and K. Nahrstedt, "ACLA: A framework for access control list (ACL) analysis and optimization," Communications and Multimedia Security Issues of the New Century: Springer, pp. 197-211, 2001.

[10] E. S. Al-Shaer and H. H. Hamed, "Modeling and management of firewall policies," IEEE Transactions on Network and Service Management, vol. 1, pp.2-10, 2004.

[11] E. Al-Shaer, H. Hamed, R. Boutaba and M. Hasan, "Conflict classification and analysis of distributed firewall policies," IEEE journal on Selected Areas in Communications, vol. 23, pp. 2069-2084, 2005.

[12] M. G. Gouda and X. Y. Liu, "Firewall design: Consistency, completeness, and compactness," Distributed Computing Systems, 2004 Proceedings 24th International Conference on, pp. 320-327, 2004.

[13] M. G. Gouda and A. X. Liu, "Structured firewall design," Computer Networks, vol. 51, pp. 1106-1120, 2007.

[14] H. H. Hamed, A. El-Atawy and E.Al-Shaer, "Adaptive Statistical Optimization Techniques for Firewall Packet Filtering," INFOCOM, vol. 6, pp. 1-12, 2006.

[15] L. Yuan, H. Chen, J. Mai, C. N. Chuah, Z. D. Su and P. Mohapatra, "Fireman: A toolkit for firewall modeling and analysis," 2006 IEEE Symposium on Security and Privacy, pp. 15-213, 2006.

[16] A. El-Atawy, T. Samak, E, Al-Shaer and H. Li, "Using online traffic statistical matching for optimizing packet filtering performance," IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications, pp. 866-874, 2007.

[17] A. X. Liu and M. G. Gouda, "Diverse firewall design," IEEE Transactions on Parallel and Distributed Systems, vol. 19, pp. 1237-1251, 2008.

[18] G. Misherghi, L. Yuan, Z. Su, C. N. Chuah and H. Chen, "A general framework for benchmarking firewall optimization techniques," IEEE Transactions on Network and Service Management, vol. 5, pp. 227-238, 2008.

[19] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach and T. Berners-Lee, "Hypertext transfer protocol--HTTP/1.1," 1999.

[20] V. G. Abhaya, Z. Tari, P. Zeephongsekul and A. Y. Zomaya, "Performance analysis of EDF scheduling in a multi-priority preemptive M/G/1 queue," IEEE Transactions on Parallel and Distributed Systems, vol. 25, pp. 2149-2158, 2014.