REGULARLY EXPECTED REFERENCE-TIME AS A METRIC OF WEB CACHE REPLACEMENT POLICY

Agung Sediyono

Universitas Trisakti, Indonesia, agung@trisakti.ac.id

ABSTRACT. The growth of Internet access was increasing significantly. In facts, more than one user access the same object so there is an opportunity to reduce this redundancy by placing an intermediate storage called cache. By this approach, the bandwidth consumption and response time of system in term of user perception can be improved. When the size of web cache is limited, it needs to manage the objects in web cache so that the hit ratio and byte hit ratio are maximized. Based on previous research the performance of cache replacement is dependent on the user/program access behavior. Therefore, the success of IRT implementation in memory cache replacement is not guaranteed a same result for web cache environment. Researcher has explored the regularity of user access and used this characteristic to be included in a metric of web cache replacement. Other researcher uses the regularity to predict the next occurrences and combine with past frequency occurrences. In predicting process, they use statistic or data mining approach. However, it takes time in computing prediction process. Therefore, this paper proposes a simple approach in predicting the next object reference. This approach is based on assumption that the object could be accessed by user regularly such DA-IRT that be used to calculate the time of next object reference called the regularly expected reference time (RERT). The object with longer RERT will be evicted sooner from the web cache. Based on experiment result, the performance of RERT is dependent on user access behavior and opposite of DA-IRT policy.

Keywords: inter-reference time, expected reference time, web cache replacement

INTRODUCTION

The growth of Internet access was increasing significantly. In facts, more than one user access the same object so there is an opportunity to reduce this redundancy by placing an intermediate storage called cache. By this approach, the bandwidth consumption and the response time of system in term of user perception can be improved. In case of limited size of web cache, it needs to manage objects that are saved in the web cache. The objects in web cache are managed by saving only the valuable objects so that the hit ratio (HR) and the byte hit ratio (BHR) can be maximized. In determining the valuable objects, researchers propose several approaches such as based on popularity, size of object, cost of bandwidth, response time, inter reference time, or combining those metrics. Based on the previous researches, it can be concluded that there is no approach being suitable for all organization. On the other hand, the performance of cache management is depended on the user behavior. By knowing the user behavior, it can be expected the next object requested by user. The user behavior can be learned by exploring the workload of web cache. Therefore, researcher focus on studying the web cache workload for several organization. The research of workload of the web cache was conducted extensively (Breaslau et al., 1999; Cohen&Kaplan, 2008; Sediyono, 2008).

Breaslau, et al. (1999) concluded that the distribution of the web requests follow a Zipf-like distribution and this model can explain why the performance of the web cache is certain asymptotic properties. Then, Cohen&Kaplan (2008) measured the regularity of the workload and use it to design the optimal cache replacement algorithm. Meanwhile, Sediyono (2008) reveals the relation between inter-reference time (IRT) and popularity showing the strong relationship and regularity. Based on this fact, there is an opportunity to implement IRT and its regularity in evicting objects from the web cache. Object with longer dynamic-average of inter-reference time (DA-IRT) is lesser popular and will be evicted from the web cache. By this policy, the performance of web cache outperforms LFU policy for certain log traces. On the other hand, it is also dependent on user behavior as previous research (Sediyono, 2009). By recording the state of purged objects, the performance of web cache can be improved significantly especially for log trace where the DA-IRT has a poor performance. However, it needs an additional storage to save the state of purged object and extra process in finding information of purged object (Sediyono, 2009.a). This paper tries to implement inter-reference time metric in the other side that is to uses the regularity of user access behavior. Based on the fact that web objects are accessed regularly by the users (Cohen&Kaplan, 2008; Sediyono, 2008), it can be predicted the next object that will be referenced by the users. The parameter of prediction is combined with parameter of passed access to be a metric for evicting object from web cache. (Yang&Zhang, 2003; Songwattana, 2008). However, it takes time in computing prediction process. Therefore, this paper proposes to predict the next object by the regularly expected reference time (RERT). RERT is calculated from the time difference between regular reference time and current access time of accessed object.

RELATED WORK

Inter-reference time of the successive object requests was extensively discussed and implemented in memory cache replacement (Phalke&Gopinath, 1995; Jiang&Zhang, n.d; Takagi&Hiraki, 2004). Phalke&Gopinath (1995) explored the behavior of inter-reference gap (IRG) that is the time interval between successive references to the same address. They concluded that the IRG has, in general, a repetitive behavior. Therefore, they applied a k order Markov chain to predict the next reference in the future. Based on the experiment, this method can improve the cache replacement until 37% over the Least Recently Usage (LRU). Jiang&Song (Jiang&Zhang, n.d) introduce the LIRS cache replacement based on Inter-reference Recency (IRR) Set. IRR uses the number of references of the other objects that is in the inter-reference time of certain object. On the other hand, they use spatial locality instead of temporal locality. They argue that the age of the object in the cache can be measured by counting the number of references of the other object after the object measured is entered into the cache. LIRS uses two blocks of cache: LIR for low inter reference and HIR for high inter reference. By using this approach that is not depending on the detectable pre-defined regularities in the reference of the workloads, LIRS can improve the LRU performance. Meanwhile, Takagi&Hiraki (2004) argue that each memory address has own IRG distribution, so that they suggest to make individual probability distribution of each memory block and use the distribution to estimate the next reference in the future. This approach depends on the historical data so that it can introduce the complexity in both memory and computation.

Even though IRT has extensively discussed and implemented successfully in memory cache replacement, the research on the inter-reference time for web cache replacement was rarely conducted. Sediyono (2008) tries to explore the relation between IRT and object popularity. It can be conclude that there is a strong correlation between those parameters. On the other word, it is reasonable to implement IRT in web cache environment. For efficient computation, instead of using IR, the dynamic average of inter-reference time (DA-IRT) was proposed (Sediyono, 2009). DA-IRT is calculated based on step by step average of inter-reference time. By implementing DA-IRT the performance of web cache outperforms LFU for certain number of web traces. On the other word, the performance of web cache is still dependent on the user access behavior. This result is conformity to previous research (Lindermann, n.d). The extended research of Sediyono (2009) was proposed to maintain the state of purged objects. This approach was applied in order to maintain the state of popular objects that often entering and getting out from the web cache because of object replacement process. By this approach, the performance of web cache can be improved significantly, especially for poor DA-IRT performance. However, it needs extra storage and processing in maintaining the state of purged objects.

Cohen&Kaplan (1999) measured the regularity of the workload and use it to design the optimal cache replacement algorithm. Sediyono (2008) also found the regularity of user access behavior in term of IRT. The regular characteristic of user access behavior can be used for predicting next object reference based on previous information. Therefore, Yang & Zhang (2003) use data mining approach to predict the future frequency of occurrences for current object, and combine to past frequency of occurrence of current object in GDSF policy. The same approach was proposed by Songwattana (2008), he makes a prediction of future occurrences using statistic model, and then combines the probability of future frequency of object with the past frequency of object in GDSF policy. Based on the simulation, this approach outperforms the native GDSF. However, for online implementation, this approach will take time in processing a table as a reference in predicting. Therefore, this paper proposes a simple approach that uses expected time of next occurrence. Object with longest expected time will be evicted from the web cache.

REGULARLY EXPECTED REFERENCE-TIME

In this section it will be presented why RERT is chosen as a metric in evicting object in the web cache and how this metric will be implemented so that the computation time will be efficient.

Rationale

Based on DA-IRT (sediyono, 2009), an object is evicted from the web cache if the object has largest DA-IRT values. It means that DA-IRT policy assumes that object with smallest value of DA-IRT will be access soonest, so it has to be kept in the web cache. This assumes is valid if the object is access regularly. If so, it can also be viewed in opponent side that if we stand in a current time of referred object, it can be calculated how long the object will be accessed again. Because of calculating the future reference time of accessed object, this was called an expected reference time of the object. On the other word, RERT is opponent of DA-IRT as long as the objects are accessed regularly.

RERT is calculated based on DA-IRT value. DA-IRT value of object *n* accessed at time *i* is calculated using formula as in equation 1 [4]

$$IRT_{ni} = \frac{IRT_{ni-1} + (t_{ni} - t_{ni-1})}{f_{ni}} \tag{1}$$

where IRT_{ni} is a dynamic average of inter-reference time of the reference n at time ith, $t_{ni} - t_{ni-1}$ is inter-reference time of reference n at time i, f_{ni} is frequency of reference n. Notable, for the first timer object, the average of IRT is assumed equal to the first reference time and placed into the web cache based on LRU policy among the first timer object. Therefore, RERT is calculated as follows:

$$RERT_{ni} = \left(T_{nl} + mIRT_{ni}\right) - T_{nc} \tag{2}$$

where T_{nl} is the last time of accessed object *n*, T_{nc} is current time of accessed object *n*, and m is integer value so that $RERT_{ni}$ is not lower that zero.

Object with largest RERT will be evicted first from the web cache. This approach assumes that object with smaller RERT will be accessed sooner, so that it has to be kept in web cache in order to be hit.

Implementation

The RERT is implemented using linked list data structure. The data attribute saved in the web cache are object size, lastly referred time, cumulative IRT, and frequency of reference. The cumulative IRT and frequency of reference of the object will be updated if the object is referred. The algorithms of the RERT as follows:

```
Input: X the object requested by user

Process:
If  Object X is in Cache
    Add cumulative IRT by current IRT
    Increase the frequency by one
    Calculate  DA-IRT value
    Calculate HR, BHR
Else
    While there is no the space of cache for X
        Calculate RERT and evict the object with largest RERT value
    Add object X into the cache using LRU policy among the first timer

Ouput: HR, BHR
```

METHODOLOGY

The methodology used in this research is experimental-based methodology. The experiment is conducted by simulating the web cache replacement policy and using the web trace log as a input. This section describes and discusses about the evaluation criteria and the data preparation for the web cache simulation.

Evaluation Criteria

The criteria of evaluation are determined to assess the performance among web cache replacement policies. Based on the previous research, the criteria of evaluation for the web cache replacement are Hit Ration (HR) and Byte Hit Ratio (BHR). The HR is ratio between the number of references and number of requests. Meanwhile BHR is ratio between the number of byte of the references and the number of byte of the requests. The formulation of HR and BHR as follows:

$$HR = \frac{\sum hit}{\sum request} \tag{3}$$

$$HBR = \frac{\sum byte\_hit}{\sum byte\_requested} \tag{4}$$

Data Preparation

This section discusses about data testbed for simulation beginning from the raw data, data processing, and data properties.

The raw data for the experiment are collected from three companies: Garuda Indonesia Airways (GIA), PT Telkom (Telkom), and PT Peti Kemas (PetiKemas). The GIA web caches have been collected as long as three weeks from November, 1st till 18th 2008, and the Telcom web caches have been collected for one week from Nopember, 2nd till 8th 2008. Meanwhile, PetiKemas web cache has been collected for five weeks from June, 26th 2008 till July, 31th 2008.

Before the web caches workload is explored, the web caches are filtered so that only the cacheable object that will be explored. To filter the cacheable objects, this paper adopts the rule that was also used by Casilari&Trivino-Cabrera (2008). The rule is the web request that contain the '?', 'cgi', or 'cgi-bin' will be discarded from the web cache log, and only those request with a cacheable response code, that is, 200 (OK), 203 (Partial), 206 (Partial Content), 300 (Multiple Choices), 301 (Move), 302 (Redirect), and 304 (Not Modified) will be used in the experiment.

Simulation

The simulation is conducted using computer program in C# language. The web cache replacement policies that are compared in the experiment are LRU, LFU, GDS(1), DA-IRT, and RERT itself. The size of web cache is varied in range 20, 40, 60, 80 percent of the total size of distinct requests in testbed. These policies are implemented in same testbed and then HR and BHR of each web cache replacement policy and web cache size are calculated.

Table 1. The properties of the web caches under investigation

	PT Garuda Indonesia Airways (1-18 Nov 2008)				PT Telkom (2-8 Nov 2008)			Peti Kemas (26 Juni - 31 July 2008)
	GIA #1	GIA #2	GIA #3	GIA #4	Telcom #1	Telcom #2	Telcom #3	
# of Request	3,544,156	8,269,922	4,717,459	3,137,920	5,014,879	4,560,189	8,219,840	7,558,496
# of Cachable Request	1,372,801	3,195,265	2,194,430	1,664,758	2,208,864	1,385,718	3,519,394	3,253,394
Request rate daily	**76,266**	**177,514**	**121,912**	**92,486**	**315,552**	**197,959**	**502,770**	**92,954**
% of Cachable Request	**38.73**	**38.64**	**46.52**	**53.05**	**44.05**	**30.39**	**42.82**	**43.04**
Total Size of Cachable Object (MB)	13,957.7	37,774.8	16,557.4	31,971.9	245,888. 8	99,323.9	245,888.8	66,548.2
One Timer	298,121	649,826	417,580	389,405	35,024	16,584	34,894	792,099
% of One Timer	**21.72**	**20.34**	**19.03**	**23.39**	**1.59**	**1.20**	**0.99**	**24.35**
# of Distinct Request	379,746	839,080	532,241	487,728	594,061	408,016	923,313	1,030,870

EXPERIMENT RESULT AND ANALYSIS

In this analysis, it will only be discussed the experiment of GIA#1 and Telkom#2, because these testbeds have shown an extreme characteristic (Sediyono, 2009; Sediyono, 2009.a). In general, increasing web cache size tends to increases either hit ratio or byte hit ratio. This characteristic confirms to the previous research. GDS(1) outperforms all web cache replacement policies in terms of HR, but it has a poor performance in terms of BHR. This result is also conformity with previous research. As predicted before, the characteristic of RERT is opposite of DA-IRT. This conclusion is obtained from the HR characteristic of RERT and DA-IRT in Figure 1 and 2. On the other word as in previous research, RERT is also dependent on the user behavior in accessing Internet. Therefore, it needs to recognize another metric that can be used as a reference in switching among replacement policies, so that the system can adapt the different user behavior.

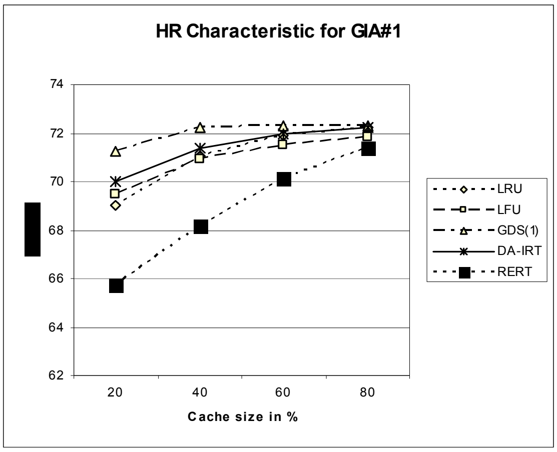

Figure 1. HR Characteristic of GIA #1

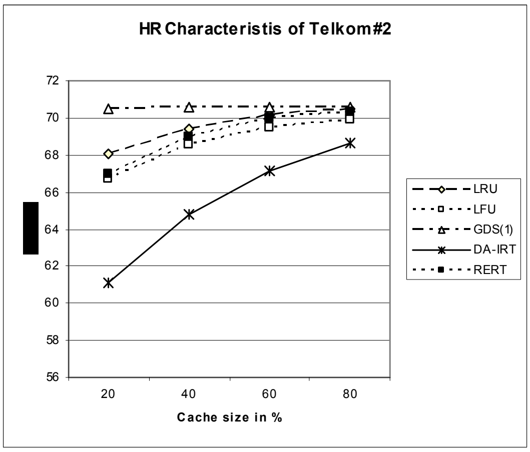

Figure 2. HR Characteristic of Telkom #2

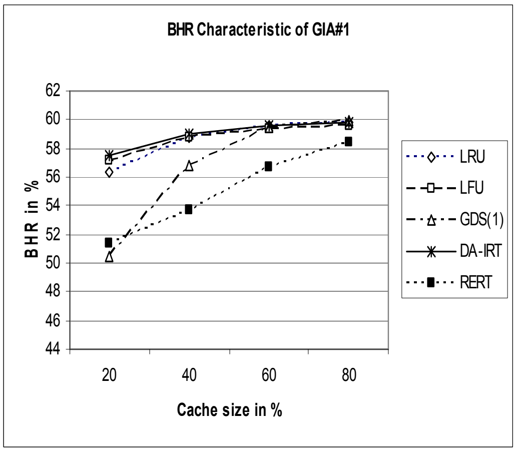

Figure 3. BHR Characteristic of GIA#1

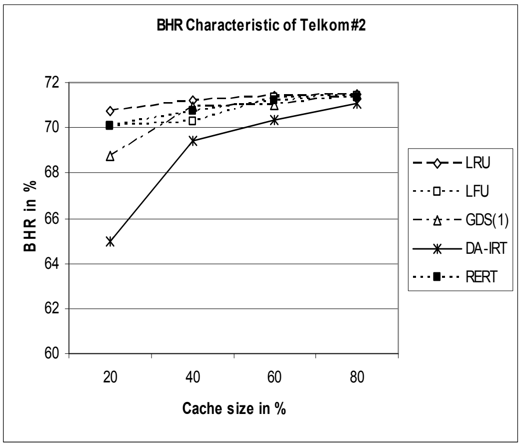

Figure 4. BHR Characteristic of Telkom #2

CONCLUSION AND FUTURE WORK

Based on the simulation result from two testbeds, it can be concluded that the performance of RERT is dependent on the characteristic of workload. Moreover, the characteristic of RERT is opposite of DA-IRT in term of HR. Therefore, if it can be recognized the characteristic of workload so that the system can determine which one of policy that will be implemented, the web cache performance can outperform for all testbeds.

Based on the simulation result in current research, it needs to find the web cache replacement policy that can adapt the difference workload characteristic. Using this approach, it can be expected that system can learn and adapt the user behavior so that the performance of web cache can be optimized.

ACKNOWLEDGMENT

Special thanks to Garuda Indonesia Airways, PT Telkom, and PT Peti Kemas that have prepared and given the trace log of the web cache for this research.

REFERENCES

Lindemann, Cristoph & Waldhosrt, O.P. (n.d). Evaluating the impact of different document types on the performance of web cache replacement schemes. http://www4.cs.uni-dortmund.de/~Lindemann/

Breaslau, Lee et al. (1999). Web caching and zip-like distributions: evidence and implications. IEEE INFOCOM, Vol XX No. Y.

Casilari, F.J. & Trivino-Cabrera, A. (2008). A windows based web cache simulator tool. Conference of SIMUT Tools, Marsielle, France.

Cohen, Edith & Kaplan, Haim. (1999). Exploiting regularities in web traffic patterns for cache replacement. STOC'99 Atlanta GA, USA

Jiang, Song & Zhang, Xiaodong. (n.d.). LISRS: an efficient low Inter-reference recency set replacement policy to improve buffer cache performance. IEEE explorer.

Phalke, Vidyadhar & Gopinath, Bhaskarpillai. (1995). An inter-reference Gap model for temporal locality in program behavior. SIGMETRICS'95 Ottawa, Ontario, Canada.

Sediyono, Agung. (2008). Exploiting Inter-reference Time Characteristic of Web Cache Workload for Web Cache Replacement Design. Proceeding of ICOCI Langkawi Malaysia.

Sediyono, Agung. (2009). Dynamic average of inter-reference time as a metric of web cache replacement policy. Proceeding of International Conference on Rural Information and Communication Technology 2009. ITB Bandung. ISBN 978-979-15509-4-9.

Sediyono, Agung. (2009.a). Perfectly Dynamic average of inter-reference time as a metric of web cache replacement policy. Proceeding of International Conference on ICTS ITS, Surabaya.

Songwattana, Areerat.(2008). Mining Web logs for Prediction in Prefetching and Caching. Third 2008 International Conference on Convergence and Hybrid Information Technology.

Takagi, Masamichi & Hiraki, Kei. (2004). Inter-reference Gap Distribution replacement: an improved replacement algorithm for set-associative caches. ICS'04 Saint Malo, France.

Yang,Q. and Zhang, H. (2003). Web-Log Mining for Predictive Web Caching. IEEE Transaction on Knowledge and Data Engineering, Vol. 15, No. 4, July/August.