Spider web inspired Dynamic Memory Allocation

Siddharth Gurdasani, Wenzhu Liu

Introduction

Spiders in particular orb web spiders, utilize the threads as an external sensory organs to learn about the spatial structure of the environment i.e., where the prey will be falling. And based on this it performs the smart move of thread pulling using its legs (as it is sitting in the center of the web) which in turn creates tension in the thread and aggravates the vibration transmission capacity of the thread. In this way, spiders which are classified into sit and wait predators optimize their prey detection efficiency and adapt according to the environment.

Spatial learning performed by the spider can be applied to optimize the dynamic memory allocation algorithms like BBSLR, in order to utilize the resources efficiently and be able to serve maximum number of simultaneous clients.

Dynamic Memory Allocation

Cache memory is a smaller and faster memory which stores the data repeatedly requested by the clients. Searching data in cache memory is time saving because of small size of the memory.

Static memory allocation works on the principle of allocating fixed amount of cache memory to the request by the client which is generally a minimum sized one. While dynamic memory allocation, considers the memory requirement of the clients and reclaims the rest of the memory (when not in use). So dynamic memory allocation calculates the cache memory based on the need of the data at run time and also manages memory more efficiently.

BBSLR Algorithm

A novel algorithm proposed by Li et al (2007), Balanced Buffer Sharing of Limited Resources (BBSLR) allocates memory dynamically to the clients based on available cache and main memory. It also adjusts the value of dm (refer to the equation) at run time according to the request distribution of the clients and available memory. In this way, it manages memory more efficiently and also maximizes the number of simultaneous clients.

It also reduces average start up delay (response time of the server) by caching the data initially, so that more clients can be served. It further reclaims the cache memory when no longer in use or the service is terminated and thus maintains equilibrium between the main memory resources and cache memory resources. So this algorithm would induce even consumption of memory from both the resources.

$$d_m = \begin{cases} \left\lceil \frac{B_i'}{B'} M' \right\rceil & (B' \neq 0 and B_i' \leq B') \\ Min\left(M', \left\lceil 2\frac{M-M'}{n} \right\rceil\right) & (B' = 0 or B_i' > B') \end{cases}$$
$$(1 \leq i \leq n)$$

M-Size of the whole memory (byte),
B-Size of the whole disk bandwidth (bps),
M'-Size of the currently available memory (byte),
B'-Size of the currently available disk bandwidth (bps),
Mi-Size of the cache used by the ith request (byte),
Bi-Size of the bandwidth used by the ith request (bps),
M_i'-Size of the cache required by the ith request served by disk stream(byte),
B_i'- Size of the bandwidth required by the ith request served by disk stream (byte) ,
n-Number of user requests in service at some time,
dm-Limit of interval cache size between two adjacent displays referencing the same video (byte)

Dynamic memory Allocation vs. Spider

As the BBSLR allocates memory dynamically depending on the request distribution and available memory, spider also varies tension levels on the radial threads according to the frequency of preys falling on either side of the web.

In the Table 1, we have demonstrated the correlation between the static model (Static memory allocation) and dynamic model (dynamic memory allocation) inspired by the spider, where the requests by the clients can be compared to the falling preys and accordingly, the memory allocated can be compared to tension levels of the threads.

In the Table 2, we are explaining the time difference in searching from the main memory and the cache memory. Also, the cache memory is changing dynamically and so some of the datum not requested by the clients is reclaimed. Due to this, there is an equal consumption of both main memory and cache memory. So due to reclamation of memory from cache, it can simultaneously serve more clients as well as can store more data and eventually lead to faster searches. This makes the server client network robust .

Results

Static Model			Dynamic model inspired by spider		
Clients	Requests	Memory allocated	Clients	Requests	Memory allocated
Client 1	1	5	Client 1	1	1
Client 2	1	5	Client 2	1	1
Client 3	4	5	Client 3	4	5
Client 4	8	5	Client 4	8	9
Client 5	4	5	Client 5	4	5
Client 6	6	5	Client 6	6	7
Client 7	7	5	Client 7	7	8
Client 8	7	5	Client 8	7	8
Client 9	2	5	Client 9	2	2
Client 10	4	5	Client 10	4	5

Table 1 : Correlation between static model and dynamic model

No	Time to load from main Memory(seconds)	Time to load from Cache Memory(seconds)
1	0.0000028610	0.0000019073
2	0.0000030994	0.0000028610
3	0.0000040531	0.0000009537
4	0.0000030994	0.0000019073
5	0.0000030994	0.0000011921

Table 2 : Dynamic allocation of cache memory

Conclusion

As the spider performs spatial learning of the environment, server client request management system can be optimized based on the request distribution of the clients. Moreover, server can dynamically allocate cache memory like the spider regulates tension on threads, based on the frequency of requests by each client. Also server can utilize the resources efficiently by providing more privileges to high requesting clients and utilize reclamation algorithms to clean up the cache memory. In this way, server memory load can be reduced and it can utilize resources efficiently.

Works Cited

- Kaihui Li; Yuanhai Zhang; Jin Xu; Changqiao Xu, "Dynamic Memory Allocation and Data Sharing Schedule in Media Server," Multimedia and Expo, 2007 IEEE International Conference on , vol., no., pp.72,75, 2-5 July 2007
- Kensuke Nakata. Spatial learning affects thread tension control in orb-web spiders. Biology Letters, 9(4), 2013.
- Jacquelyn M Zevenbergen, Nicole K Schneider, and Todd A Blackledge. Fine dining or fortress? Functional shifts in spider web architecture by the western black widow Latrodectus hesperus. Animal Behaviour, 76(3):823–829, 2008.