A Dynamic Equilibrium View of Caching Systems

F. Díaz-del-Río, J.L. Sevillano, D. Cascado.
ETS Ingenieria Informatica. University of Seville, Spain.
{fdiaz, sevi, danic}@atc.us.es.

J.M. Rodríguez-Corral.
Escuela Superior de Ingenieria. Universidad de Cadiz, Spain.
josemaria.rodriguez@uca.es

***Abstract*— In this paper, we present a simple analytical study of caching systems based on the idea of dynamic equilibrium of cache blocks, assuming the Independent Reference Model (IRM) of references. This method allows us to obtain simple closed-form expressions for parameters that are usually excluded from cache studies, such as the mean number of reads or writes per cache block while in the cache. We finally present some simulation results in order to validate the analysis.**

Keywords*— *Cache analysis, Dynamic Equilibrium, Cache Write Policies.

I. INTRODUCTION

In order to obtain simple models of the caching process, the *Independent Reference Model* (IRM), which assumes that the probability of referencing each block is constant over time, could be used [1,2]. This simplifying assumption provides tractable analyses and results with a relatively intuitive form. Although it does not consider the temporal aspects of the application-specific access patterns, it may be a useful reference model in many situations [3].

In this paper, we present a simple analysis of a cache system based on the idea of dynamic equilibrium of cache blocks, assuming the IRM. Our model includes some interesting aspects that are usually excluded from cache studies, such as write accesses (with a few exceptions such as [4,5]) and particularly write miss policies [6]. These policies can be simply reduced to two main alternatives: *write allocate-WA*, if the block is loaded on a write miss, and *no-write allocate-NWA*, which in case of a write miss the block is not loaded into the cache. Through some simulation results we will show that the estimation using our analysis is acceptable for many configurations.

II. ANALYSIS.

First, let us consider a WA cache with l lines or blocks. Let us denote M_R and H_R as the miss and hit rates, respectively. As we stated above, we assume the IRM, that is, the probability of accessing each (memory) block is constant over time, but not necessarily with the same probability. Thus we are considering the general case of non-uniform IRM [1]. We also assume a random block replacement algorithm where appropriate. We suppose that all the other design alternatives are included in the model through parameters like the miss rate. Finally, the mean percentage of read and write accesses are F_R and F_W, respectively.

In this work we define a new parameter: the mean number of accesses per block. This parameter gives us an intuitive idea of the locality of memory references, and allows an insight interpretation of the caching process, as well as a simple estimation of other useful parameters like the percentage of dirty blocks (see [3]).

Let n_R (n_W) be the mean number of reads (writes) into a single block, that is, the total number of reads (writes) into the cache divided by the number of cache blocks l ($n_W=N_W/l$, $n_R=N_R/l$). The value of n_W after a new access will be:

$$n_W = F_W\left(H_R n_{W,wh} + M_R n_{W,wm}\right) + F_R\left(H_R n_{W,rh} + M_R n_{W,rm}\right)$$

where $n_{W,wh}$ is the new value for n_W after a write hit, $n_{W,wm}$ the new value for n_W after a write miss and so on. In case of a write hit the total number of writes into the cache increases by 1. In case of a miss a block must be thrown out (with n_W write requests in the mean), and then the new block is written once. Therefore,

$$n_{W,wh} = \frac{N_W+1}{l};\quad n_{W,wm} = \frac{N_W - n_W + 1}{l}$$
$$n_{W,rh} = \frac{N_W}{l};\quad n_{W,rm} = \frac{N_W - n_W}{l}$$

As a result, in a steady-state situation we obtain

$$n_W = F_W\left(H_R\frac{l\times n_W+1}{l} + M_R\frac{l\times n_W - n_W + 1}{l}\right) + F_R\left(H_R n_W + M_R\frac{l\times n_W - n_W}{l}\right) \Rightarrow n_W = \frac{F_W}{M_R}$$

For every write access we have F_R/F_W read accesses. Then the average "span of life" of a block (the expected number of references to a block while cache resident) is the sum of the write and read accesses, which results to be the inverse of the miss rate:

$$n_{total} = n_W + n_R = \frac{F_W}{M_R} + \frac{F_W}{M_R}\frac{F_R}{F_W} = \frac{1}{M_R}$$

This result can be seen intuitively since a miss every n_{total} accesses is just the miss rate. In order to refine the model, we

can drop the assumption of a constant miss rate. Actually, the miss rate of read and write accesses may be quite different, so if we take into account this difference,

$$n_W = \frac{F_W}{F_W M_{RW} + F_R M_{RR}}; n_R = \frac{F_R}{F_W M_{RW} + F_R M_{RR}} \quad (1)$$

where M_{RW} denotes the writes miss rate, and M_{RR} the reads miss rate. Note that the denominator is an "average" miss rate. Note that the cache size (i.e., the number of blocks l) does not explicitly appear in these equations. In general all of the design parameters only influence the mean number of accesses per block insofar as they affect the miss rate.

These results can be interpreted as follows. In a WA cache, a block replacement occurs whenever a read or write access misses. So, the "average" miss rate is the probability that a line is loaded into the cache, that is, it can be considered as a measure of how many lines have been loaded into the cache. As a result, equation 1 can be interpreted as the total number of writes (reads) divided by the total number of blocks that have been loaded into the cache, so they provide the mean number of writes (reads) per block while in the cache.

Now, let us consider a NWA cache. The only difference is that a write miss does not change n_W because the data is modified only in the lower level. Thus,

$$n_W = F_W\left(H_{RW}\frac{l\times n_W + 1}{l} + M_{RW} n_W\right) +$$
$$F_R\left(H_{RR} n_W + M_{RR}\frac{l\times n_W - n_W}{l}\right) \Rightarrow n_W = \frac{F_W H_{RW}}{F_R M_{RR}}$$

The numerator becomes the percentage of successful write accesses and the denominator the percentage of unsuccessful read accesses. In a NWA cache a block replacement only occurs with a read miss, so $F_R M_{RR}$ is a measure of the number of blocks that have been loaded into the cache. Among these blocks, only those where a write access has hit will be modified ($F_W H_{RW}$). Similarly, the mean number of read accesses becomes

$$n_R = F_W\left(H_{RW} n_R + M_{RW} n_R\right) +$$
$$F_R\left(H_{RR}\frac{l\times n_R + 1}{l} + M_{RR}\frac{l\times n_R - n_R + 1}{l}\right) = \frac{1}{M_{RR}}$$

An interpretation of this result is that, since a block replacement only occurs with a read miss (NWA cache), a miss every n_R read accesses gives us the reads miss rate.

III. Simulation

In order to validate this analysis, a trace-driven simulator has been developed. We performed several simulation experiments using different statistically generated IRM traces and the results are always very close to those of the analysis. As an example, we present the results for a configuration similar to one of those used in [1]. We assume a memory size of 16K, with 3 "segments" so that each position of a segment have the same probability of access. A fraction of 9/16 of the accesses are references to one of the first 1K positions, 3/16 to the following 3K positions and 4/16 for the rest 12K positions. We present the mean number of writes (Fig. 1a) and reads (Fig. 1b) per block while in the cache for different configurations. Configurations are shown using the following notation: first, the cache size and then the block size indicated as the exponent of a power of two. For instance, 10:3 means a cache of 2^{10}=1K blocks of 2^3=8 bytes each one.

Even for this "highly" non-uniform access pattern, and also for cache sizes of up to 50% of the memory size, simulation results are very close to our analysis. Simulations using real CPU data traces show that the above expressions provide an accurate estimation, with the difference always under 14% even n_W and n_R have high values [3].

IV. Conclusions

The use of a model for cache systems based on the concept of dynamic equilibrium, and assuming the independent reference model, allows us to obtain closed form equations for some parameters such as the mean number of reads or writes per cache block. These closed form expressions allow an intuitive interpretation of the caching process, as well as a simple estimation of other useful parameters like the percentage of dirty blocks (which permits to estimate the mean write traffic between the cache and the following storage level, see [3]). Future work will address the extension of this model to other useful parameters and to more accurate reference models.

V. Acknowledgments

This work was supported by contracts TIN2006-15617-C03-03 and P06-TIC-2298.

VI. References

[1] D. Buck, M. Singhal, An Analytic Study of Caching in Computer Systems, Journal of Parallel and Distributed Computing, 32 (1996) 205–214.

[2] D. Starobinski, D. Tse, Probabilistic Methods for Web Caching. Performance Evaluation, 46 (2001) 125-137.

[3] F. Díaz del Río et al., A dynamic equilibrium based model of caching systems. Technical Report. Dep. ATC, Univ. Sevilla (2007).

[4] D.W. Miller and D.T. Harper III, Performance Analysis of Disk Cache Write Policies. Microprocessors & Microsystems, 19-3 (1995) 121-130.

[5] A.J. Smith, Characterizing the storage process and its effect on the update of main memory by write-through, Journal of the ACM 26-1, (1979) 6-27.

[6] N.P. Jouppi, Cache write policies and performance, WRL Research Report 91/12 (1991).

Figure 1a: writes

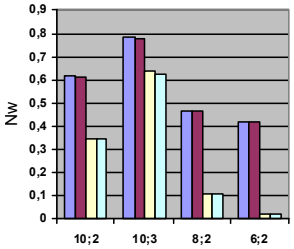

Figure 1b: reads

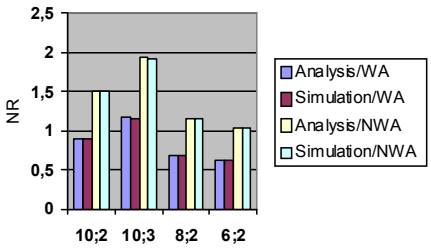

Figure 1. Mean number of writes/reads into a single block.