

The analysis of PQRS and other bias models

Double blind submission

ABSTRACT

A new family of models for I/O access patterns, the PQRS family was recently introduced and studied by Wang et. al. In particular these authors studied experimentally average seek distances and cache hit rates for these models. In this paper we compute analytically average seek distances and cache hit rates for PQRS models. We introduce an efficient, parameter independent, caching algorithm for all PQRS models and analyze it's hit ratio. The existence of such an algorithm shows that PQRS generated traces are too predictable to be used as benchmarks. We show how the models can be amended to eliminate predictability. We also introduce the, parameter dependent, optimal caching algorithm for PQRS models and compute it's hit ratio. We show that models with time independent spatial distributions produce the lowest hit ratios. We compute the average seek distance and show that the models whose spatial distribution is time independent have the largest average seek distance. thus proving a fact which was previously noted experimentally. We also show that the output stream of read misses coming from a PQRS input stream after passing through a cache is also approximated by a PQRS model. The results also quantify and clarify the relations between the various entropies of the models and properties such as burstiness and cache hit ratio. Taken together the results present a comprehensive picture of the behavior of PQRS models.

Our formulas for the average seek and for hit ratios make PQRS models much more useful in optimization applications. The predictability reducing mechanisms make them eligible for use in benchmarks. In addition we introduce a larger class of models which we call bias models. We extend many of our results to this larger class. We then provide evidence that this larger class is needed for more realistic modeling of real workloads.

1. INTRODUCTION

For the last four decades researchers have attempted to

model internal and external memory I/O access patterns. I/O access patterns are in general very diverse, hence, no single family of models can be expected to model well all I/O access patterns. Instead the goal is (should be) to produce models which can capture various spatial and temporal characteristics of I/O activity. Continuous research has yielded many interesting classes of models each with it's own set of appealing properties. Several early models include the random reference model ([6]), the sequential reference model ([6]), independent reference model a.k.a the IRM ([6], [2], [1], [15]), partial Markov and Markov models ([1]) and spatio-temporal renewal process models ([1], [11]). A newer example is the phased workload model of [5] which addresses dependencies between workloads sharing the same resource. It has been useful in some applications such as data configuration design [3].

These models are largely based on classical stochastic processes. They are very effective in capturing some of the spatial characteristics of I/O access patterns, however, they are mostly static in nature and do not capture well burstiness.

Recently an interesting new family of models, the PQRS models, was introduced by Wang et. al [14]. PQRS models are not based on classical stochastic processes, in fact, they correspond to singular, self similar, measures on the unit square. Such measures have been studied mathematically by several authors, [8],[9],[10], however, they have previously not been suggested in the context of modeling I/O patterns. The analysis of these measures as models for I/O patterns motivates some problems which are different in nature than the ones which were studied previously by mathematicians.

An interesting feature of the PQRS models is that they do not produce I/O as a time series, that is in chronological sequence. Instead the model chooses many I/O addresses and access times independently and then reorders them chronologically. In addition the models are generally based on singular measures of the unit square, meaning that some combinations of addresses and access times are far more likely than others. This latter property is responsible for the bursty, space-time localized, nature of the I/O access patterns produced by the model and the strong spatio-temporal dependencies that the pattern displays. These features closely mimic the observed behavior of many I/O access patterns. Another feature of these models is that they require very few parameters which are easily extractable from a sample I/O trace.

Wang et. al used the model to produce synthetic traces which closely resembled some real traces that they have examined. They computed experimentally hit ratios and av-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright 2001 ACM X-XXXXX-XX-X/XX/XX ...\$5.00.

erage seek distances for the model based traces and showed that the numbers matched well with those of the real traces.

Wang et. al. also observed experimentally that when restricting themselves to a subclass of the PQRS models which they called I-models they obtained average seek distances which were too high and hit ratio's which were too low. Intuitively I-models are PQRS models which lack space-time correlations.

In this paper we

1) Enlarge the family of PQRS models by introducing the family of bias models.

2) Present and analyze the optimal cache algorithm for PQRS models.

3) Present for the family of PQRS models a universal, parameter independent, caching algorithm whose asymptotic hit ratio (as the size of the data set grows) tends to 1 for all models in the family. Furthermore the algorithm is static in the sense that it places in cache a predetermined set of addresses and never changes the content of the cache. The existence of such an algorithm indicates that PQRS models are too predictable.

4) compute analytically the average seek distance for PQRS models and more generally for bias models.

5) Show that the output stream of cache misses coming from a PQRS input stream resembles a PQRS model whose parameters depend on the parameters of the input stream and the size of the cache. In particular we show how to compute the average seek distance of the output stream.

6) Show how to ammend PQRS (or bias) models so that they become less predictable and hence suitable for benchmarking purposes.

7) Prove that I-models are extremal within the class of bias models in the sense that they yield the largest average seek distances and the lowest hit ratios. These results provide a theoretical justification for facts which were observed experimentally in [14]

Overall our results provide a comprehensive understanding of the features and problems associated with PQRS like models.

In addition the analytic computations of average seek distances and hit ratios greatly enhance the aplicability of PQRS or bias models in self tuning optimization tools. Such tools may need to repeat such calculations a very large number of times, [3]. Using simulations to experimentally determine these quantities would be a prohibitively slow process.

The results of this paper further suggest that binary bias models can be rather safely used in estimating seek distances, while they should be treated with more caution when it comes to estimating hit ratios. All bias models will produce hit ratios close to 1, hence, it is difficult to model relatively random workloads using these models.

As an example we will shortly explain why essentially all the models treated in this paper have very efficient cache algorithms. On the other hand when biases predicted by these models exist they can be exploited in designing new cache

algorithms of the type presented in this paper. These algorithms are very different from the ones usually employed, like LRU and it's many variants.

To understand the conceptual difference between modeling seek distances (or response time) and hit ratios we present the following simple example which is indicative of the situation one encounters with the PQRS models.

Assume we have 1024 tracks on a disk. suppose we access these tracks using uniformly distributed and independent I/O requests. Each track is accessed with probability $1/1024$. Consider another model which accesses only tracks 0, 32,64,96,128,...,992 uniformly and independently. In terms of seek distance and response time calculations it will be very difficult to separate the two models. However we note that only 32 out of the 1024 tracks are accessed in the second case and placing all of them in cache will yield a 100 percent hit ratio using a cache whose size is 3 percent of the total data. the second model is in fact a very simple example of a binary bias model. In this case the 4 least significant digits are always set to 0. Repeating the same experiment with 2^{20} tracks and accessing only tracks divisible by $2^{10} = 1024$ we will get two models which are even more difficult to distinguish in terms of response time while the second model requires a cache of size $1/1024$ to obtain a 100 percent hit rate.

There is another indication that the hit ratio calculation is problematic. Consider again the disk with 1024 tracks. Consider now a model which uniformly and randomly accesses tracks 1,33,65,97,129,...,993. In terms of response time and disk behavior this model is essentially indistinguishable from the model which accesses tracks 0,32,...,992. Notice however that placing tracks 1,33,65... in cache will yield a 100 percent hit ratio for one model while yielding a 0 percent hit ratio for the second model. In the same manner tracks 32,64,... which were previously so effective in cache are useless now. All this comes to show that hit ratio is a very model sensitive performance metric.

While the mathematical results which quantify the corresponding calculations for the PQRS model are a bit more technically demanding they essentially capture the same heuristic behavior as in the example.

In the paper we will not attempt to fully pursue the computations to their most accurate form, but rather, we will choose varying degrees of accuracy in the hope of presenting the main mathematical techniques and results which can be used to obtain even more accurate results if desired.

The paper is organized as follows.

In section 2 we briefly review the definition and basic properties of the PQRS models as introduced in [14] and define bias models as a natural generalization.

In section 3 and 4 which form the technical core of the paper we consider seek distance and hit ratio computations for bias models.

In section 5 we present a discussion on the implications of our results and in section 6 present some statistics which we have gathered on the existence of binary biases in address bits of real traces.

2. BIAS MODELS

2.1 informal definition and discussion

A binary bias model is a model which assumes that some of the bits in the addresses of I/O requests and their ar-

rival times are statistically biased and possibly correlated. A PQRS model is a binary bias model in which the biases and correlations are the same for all the bits. We also call such models self similar since the bit statistics are the same for the more significant bits and the less significant ones, hence the statistics behave the same at all levels of granularity. We can understand the complete behavior of a self similar model by looking at a very small range of addresses for a very short period of time, hence self similar models are highly predictable. General bias models are also predictable but to a lesser extent.

In the same way we may define a k-ary bias model to be a model which assumes that the digits of the I/O request address and arrival time, written in base k are biased and/or correlated.

Correlating the statistical bias of the bits (digits) of the address and time stamp of an I/O means that the spatial distribution of requests will vary with time, often radically so. Symmetrically, the distribution of times in which various addresses are requested varies from address to address. In many cases it is interesting to consider the overall spatial distribution of requests over all times or the overall distribution of access times over all addresses. These distributions are called the marginal spatial and marginal temporal distributions respectively. When the bias of the address bits is independent of the bias in the arrival time bits then the spatial distribution of requests is time independent and therefore coincides at any given moment with the marginal distribution. At the same time the distribution of request access times will be the same for all ranges of addresses and will therefore coincide with the marginal temporal distribution. We call models for which there are no spatio-temporal correlations I-models, where the letter I stands for independence.

The overall bias of the bits and their correlations can be measured by a quantity called entropy. Low bias is associated with high entropy while highly biased models have low entropy. A comparison of the entropy of the model with the entropies of its spatial and marginal distribution yields a measure for the correlations between the spatial (address) and temporal (time stamp) aspects of requests. When the model is an I-model the total entropy of the model is the sum of the spatial and temporal entropies, if there are correlations then the total entropy is smaller than the sum.

Many of the properties of bias models, such as burstiness and hit ratios with respect to various cache algorithms are controlled by the entropy. There are certain modifications which can be performed on bias models which will leave the entropy the same. These operations thus provide us with flexibility in constructing models which share properties similar to bias models. we shall explain later on how this flexibility can be exploited. The rest of the section is devoted to a formal exposition of the above definitions and observations.

2.2 Formal definitions

A *trace* of I/O requests is a sequence consisting of pairs (s_i, t_i) , $i = 1, \dots, N$, where s_i is a storage address and t_i is the *access time* or *time stamp*, indicating the time in which the I/O request to address s_i was issued. We assume that the t_i are ordered, thus $t_i < t_{i+1}$. We let T be the time interval $[c, d]$ for which we want to generate synthetic I/O activity and we let $S = [a, b - 1]$ denote the range of storage

addresses for the trace. We assume for the sake of simplicity that $b - a$ is of the form $b - a = k^h$, for some k and h which are integers. Our models will produce pairs (s_i, t_i) where $0 \leq s_i, t_i \leq 1$. We may translate t_i into a time stamp in the time range T by considering the linear transformation $t_i \rightarrow (d - c)t_i + c$. Similarly s_i will correspond to the address $[k^h s_i] + a$, where $[x]$ denotes the integer part of x .

A *k-ary sequence of level h* is a sequence of integers

$$i = i_0, i_1, \dots, i_{h-1}$$

of length h such that $0 \leq i_l < k$ for all $0 \leq l < h$.

As usual we may identify a k-ary sequence of level h with the number $i = \sum_{l=0}^{h-1} i_l k^l$, i will obviously be in the range $0 \leq i < k^h$. We will think of a k-ary sequence of level h either as a sequence of length h or as an integer in the range $[0, k^h - 1]$ interchangeably without further mention.

A *k-ary bias model of level h* consists of the choice of h probability distributions, $P = (p_{n,m,l})$, where

$$0 \leq p_{n,m,l} \leq 1, 0 \leq n, m < k, 0 \leq l < h$$

and satisfying $\sum_{n,m} p_{n,m,l} = 1$ for all l . For given k-ary sequences i, j of level h consider the sub square $A_{i,j}$ of the unit square given by

$$A_{i,j} = \{(x, y) \mid i/k^h \leq x < (i+1)/k^h, j/k^h \leq y < (j+1)/k^h\}.$$

Such a sub square is called a *level h sub square*. Consider the probability distribution μ_P on the unit square $[0, 1] \times [0, 1]$, which assigns to $A_{i,j}$ the measure

$$\mu_P(A_{i,j}) = \prod_{l=0}^{h-1} p_{i_l, j_l, l}$$

where \prod denotes a product of numbers. Further assume that μ_P is uniform within each sub square $A_{i,j}$. These requirements completely characterize μ_P as a probability distribution.

To produce a trace of N I/O requests, the measure μ_P is sampled N times to produce N points (s'_i, t'_i) in the unit square. The N points are reordered so that the time coordinates will be in increasing order, yielding the trace (s_i, t_i) .

If $k = 2$ we say that the model is a binary bias model, if $k = 3$ ternary and so on.

While we state our results for general values of k the binary case seems to be the most relevant for modeling purposes and we shall usually restrict ourselves to that case in the proofs in order to avoid burdensome notation.

We say that a k-ary bias model is *self similar* if $p_{n,m,l}$ is independent of l in which case we can denote it by $p_{n,m}$.

The self similar binary bias models coincide with the PQRS models which were introduced in [14]. Such models are completely specified by the parameters $p_{0,0}, p_{0,1}, p_{1,0}$ and $p_{1,1}$ subject to the condition $p_{0,0} + p_{0,1} + p_{1,0} + p_{1,1} = 1$. In [14] these parameters were called respectively, p, q, r, s , hence the name of the models. We will also use the p, q, r, s notation in this specific case.

It will also be convenient for us to consider measures corresponding to the level $h = \infty$. The parameter set for such a model is an infinite collection of probability distributions $P = (p_{m,n,l})$, where l runs over non negative integers, $0 \leq m, n < k$, $p_{m,n,l} \geq 0$ and $\sum_{m,n} p_{m,n,l} = 1$ for all l . We consider the measure μ_P , which for all finite levels h , coincides with the level h measure induced by $p_{m,n,l}$, $l < h$ on the level h sub squares. It is easy to verify that the measures for the different levels are compatible, hence μ_P is well defined. It is also easy to verify that μ_P which is constructed this way is the unique measure which coincides with the level h measure on level h sub squares. Any finite

level h measure with parameters $p_{m,n,l}$, $l < h$ coincides with an infinite level measure. The parameters $p'_{m,n,l}$ for the infinite level measure are defined by $p'_{m,n,l} = p_{m,n,l}$, for $l < h$ and $p'_{m,n,l} = 1/k^2$ for $l \geq h$.

One useful fact about self similar level ∞ measures is that they are invariant under the transformation

$$(x, y) \longrightarrow (kx \bmod 1, ky \bmod 1)$$

of the unit square onto itself.

In the same way that we defined k-ary bias measures μ_P on the unit square we can define k-ary bias measures on the unit interval. The measure is specified by a set of probabilities $P = (p_{m,l})$ and the probability assigned to the subinterval $[j/k^h, (j+1)/k^h]$ is $\prod_{l=0}^{h-1} p_{j_l, l}$.

2.2.1 Marginal bias measures

A k-ary bias measure μ_P on the unit square induces spatial and temporal marginal measures μ_P^S and μ_P^T via integration. Specifically, if $[a, b]$ is a time interval we define

$$\mu_P^T([a, b]) = \mu_P(\pi_T^{-1}([a, b]))$$

where π_T denotes the projection onto the time coordinate. The definition of μ^S is similar.

It is easy to check that the space marginal of a self similar measure $p_{m,n}$ is a self similar k-ary measure on the unit interval given by $r_m = \sum_j p_{m,j}$ and similarly the time marginal is given by $s_n = \sum_i p_{i,n}$. Following [14] we say that a self similar bias model μ_P is an I -model if $\mu_P = \mu_P^T \times \mu_P^S$. This is the same as saying that there exists r_0, \dots, r_{k-1} and s_0, \dots, s_{k-1} such that $p_{m,n} = r_m s_n$.

More generally the spatial marginal of a k-ary bias model is given by the parameters $r_{m,l} = \sum_j p_{m,j,l}$ and the temporal marginal has parameters $s_{n,l} = \sum_i p_{i,n,l}$. A k-ary bias model is said to be an I -model if the associated measure μ_P is the product of it's marginal measures.

Let t be a given point in time. We obtain an induced spatial k-ary bias model for time t by restricting the binary bias model to the interval (t, s) , $0 \leq s \leq 1$. We denote this model by μ_P^t . μ_P^t is given as follows.

Assume first that h is finite and consider $t = t_0 t_1 \dots t_{h-1} t_h \dots$ the k-ary expansion of t . The k-ary bias model μ_P^t is then given by the parameters $q_{m,l} = p_{t_l, m, l}$. We note that only the first h terms of the expansion matter, hence this holds for all t in the range $I_{t,h} = \left[\frac{t h^k}{h^k}, \frac{[t h^k] + 1}{h^k} \right)$.

Recall that an *Independent reference model* (IRM) for a range of n storage addresses consists of a choice of probabilities p_1, \dots, p_n . Requests to the storage addresses are then generated independently of each other, with address i being requested with probability p_i .

The fact that μ_P^t is constant during $I_{t,h}$ and that requests in the model are chosen independently by sampling μ_P means that for the duration of $I_{t,h}$ the k-ary bias model with parameters P , is in fact an IRM whose parameters are encoded by $q_{m,l}$. We will take advantage of this fact by applying results on the average seek estimates and caching algorithms for the IRM, ([15], [4], [2]) to the k-ary bias models.

The definition of μ^t for h infinite is obtained by taking the limit of the definitions for finite h .

2.2.2 Entropy

Given a probability distribution p_1, \dots, p_n we may define the (Shannon) *entropy* of the distribution as

$$H(p_1, \dots, p_n) = - \sum_{i=1}^n p_i \log(p_i)$$

the entropy is well known to be a measure of the randomness inherent in the distribution (or the information gained by sampling it) [13]. As indicated in [14] and as we shall see more formally later on the entropy of the marginal distributions plays an important role in quantifying cache hit ratios and burstiness of a PQRS model.

More formally, given a level h bias model with parameter set P we define it's level l entropy, $H(P, l)$, as the entropy of the probability distribution it induces on the k^{2l} level l sub squares. Similarly we define the level l spatial and temporal entropies $H^S(P, l)$ and $H^T(P, l)$ as the entropies of the level l marginal distributions. The mutual information $I(P, l)$ is defined by

$$I(P, l) = H^S(P, l) + H^T(P, l) - H(P, l)$$

and is indicative of the spatio-temporal correlations, that is, the correlation between the address and access time of an I/O request in the trace. When $l = h$ we denote the various entropies by $H(P)$, $H^S(P)$, $H^T(P)$ and $I(P)$. If $p, 1-p$ is a probability distribution on two elements we denote $H(p, 1-p)$ simply as $H(p)$. Obviously $H(p) = H(1-p)$.

Given a PQRS model with parameters p, q, r, s we can easily verify that the models with permuted parameters q, p, s, r, s, r, q, p and r, s, p, q share the same marginal entropies and mutual information as the original model. More generally we may act at any level l of a k-ary bias model by row and column permutations π_l, σ_l by sending $p_{n,m,l}$ to $p_{\pi_l(n), \sigma_l(m), l}$. All the models obtained by these permuted parameters share the same marginal entropies and mutual information. We note that the models obtained by applying different permutations at different levels to a PQRS model are called *random PQRS models* in [14]. Such models are distinguished within the class of binary bias models by having entropies and mutual information which depend linearly on the level. For $k > 2$ this characterization is no longer true. The parameters of binary bias models are determined up to entropy preserving permutations by the entropies $H^S(l)$, $H^T(l)$ and $I(l)$. this fact can be used to attach a binary bias model to any given trace simply by measuring it's entropies at different levels. For $k > 2$ this method does not specify the parameters and leaves much freedom with no apparent method of pinpointing a specific solution, so the choice of specific parameters beyond the entropy restrictions might be somewhat arbitrary.

We note that we can rather easily determine wether a given trace came originally from a bias model and reconstruct it's parameters uniquely if it did.

3. ANALYSIS OF CACHING IN THE K-ARY BIAS MODELS

In this section we consider some caching algorithms for various k-ary bias models and analyze their performance. We begin with an analysis of the asymptotic hit ratio for static cache algorithms as h tends to infinity. The results show that the cache related behavior of self similar k-ary bias models is very uniform and simplistic. This suggests that this class of models is probably not rich enough to capture the cache related behavior of most traces. In the following subsection we examine more closely the hit rates associated various cache algorithms.

3.1 Asymptotic hit ratios and singular spatial marginal distributions

Consider a probability distribution μ on the interval $[0, 1]$. We say that the distribution is *singular* if for all $\varepsilon > 0$ there exist disjoint intervals $I_1(\varepsilon), I_2(\varepsilon), \dots, I_{k(\varepsilon)}(\varepsilon)$ whose sum of lengths is less than ε and for which

$$\mu(\cup I_j(\varepsilon)) > 1 - \varepsilon$$

In terms of the spatial access distribution a distribution is singular (or close to being singular) if almost all the requests are directed towards a very small subset of the addresses. A key observation for the analysis of hit ratios of generic bias models is that the spatial access distribution becomes increasingly singular as the level (size of data set) of the bias model increases. As a result the spatial marginal of their limits of level ∞ are singular, hence the spatial marginal of a generic k -ary bias model approaches a singular model in the limit as the size of the dataset increases. In terms of hit ratios the main consequence is that it is easy to design extremely simple cache algorithms whose hit ratio approaches 1 (as the size of the data increases) even for very small cache sizes.

We now explore the relation between singularity of measures and high hit ratios more closely. A caching algorithm is called *static* if it fills cache memory with a certain subset of the address space permanently. Static algorithms are obviously the simplest cache algorithms since they do not place or replace anything in cache during operation.

We note that in our setup, spatial intervals $[a, b]$ correspond to address ranges. if the level of the model is h then the corresponding address range can be normalized to be $[0, k^h - 1]$. Note that the number of addresses in the range is thus proportional to the length of the interval. If we place an address range in cache then the amount of space it occupies is proportional to the length of the interval. Given this normalization we may consider cache size as a portion of the total address space, thus when we talk about a cache of size ε we mean a cache of size εk^h where k^h is the size of the address space.

Our basic observation is that when the spatial marginal of a k -ary bias model is singular there exists a static cache algorithm whose asymptotic hit ratio (as h tends to infinity) is 1 for all cache sizes $\varepsilon > 0$.

the algorithm simply sorts the addresses by their spatial marginal probability and places the addresses with the highest probability in the cache one after the other until the cache is full. Let us call this algorithm the *greedy static algorithm*. This algorithm may be traced back to [2] where it is proved to be optimal for IRMs.

Conversely a static cache algorithm with asymptotic hit ratio of 1 provides by definition the desired sets which are witnesses to the singularity of the marginal measure hence the two notions coincide.

We specialize the discussion to PQRS models. We say that the PQRS model tends to the right if $p+r < q+s$. We say that the PQRS model tends to the left if $p+r > q+s$.

Let $Alg_{2,0}$ be the static cache algorithm which sorts the addresses in decreasing order by the number of zeroes they contain in their binary expansion and places them in cache in that order until the cache is full. Let $Alg_{2,1}$ be the algorithm which sorts the addresses in increasing order by the number of zeroes they contain in their binary expansion (equivalently in decreasing order by the number of ones) and places them in cache, in that order.

We have the following trivial observation with rather amusing consequences.

Observation: If the PQRS model tends to the right then the algorithm $Alg_{2,0}$ is the greedy static algorithm and by singularity of the spatial marginal has asymptotic hit ratio 1 for all fixed cache sizes. Similarly, if the PQRS model tends to the left then the algorithm $Alg_{2,1}$ is the greedy static algorithm with asymptotic hit ratio 1.

Note that $Alg_{2,0}$ and $Alg_{2,1}$ are parameter independent static cache algorithms meaning that they are entirely independent of workloads and access patterns. They place in cache a fixed, pre-specified set of addresses! We may combine the two algorithms into a single algorithm Alg_2 which splits the cache in two and fills each half statically according to $Alg_{2,0}$ and $Alg_{2,1}$ respectively. According to the observation Alg_2 will have asymptotic hit ratio 1 for all PQRS models with $p+r \neq q+s$.

We can similarly define $Alg_{k,n}$, $0 \leq n < k$, as the static algorithm which sorts the addresses by the number of n 's in the k -ary expansion of the address and places them in cache in descending order. The algorithm Alg_k which divides the cache into k equal parts and employs $Alg_{k,n}$ on the n 'th part has asymptotic hit ratio 1 for all self similar k -ary bias models whose spatial marginal is not uniform.

Remark: we may combine the algorithms Alg_k for all k to obtain a single static cache algorithm Alg_{SF} (SF stands for self similar) with asymptotic hit ratio 1 for all self similar k -ary bias models with non uniform spatial marginal. The algorithm orders all pairs (k, n) as an infinite sequence, first by k and then by n , thus the sequence starts with

$$(2, 0), (2, 1), (3, 0), (3, 1), (3, 2), (4, 0) \dots$$

the cache is divided into, say, h equal size pieces. We employ $Alg_{k,n}$ for the first h sequence pairs (k, n) on the different pieces. As will be seen in the next subsection Alg_k requires an exponentially (in h) small cache to obtain an asymptotic hit ratio of 1. For a cache of fixed size $\varepsilon > 0$, Alg_k , for any k , will asymptotically have a cache area of size ε/h to work with and hence Alg_{SF} achieve asymptotic hit ratio 1 for any self similar bias model with non uniform spatial marginal.

The procedure we just described is the static cache algorithm analogue of the basic fact that a union of countably many measure zero sets has measure zero.

3.2 Computing hit ratios

We quantify the performance of the preceding cache algorithms. As before we will concentrate on the binary case for simplicity.

We let $L = p+r$ and $R = q+s$ be the probabilities of going left and right in the PQRS model. Assume $L > R$. For a pair of non negative integers $n \geq m$ we let

$$B(n, m) = \frac{n!}{m!(n-m)!}$$

denote the corresponding binomial coefficient. Let c be such that $Lh - c\sqrt{LRh}$ is an integer. We first quantify the size of the cache needed by $Alg_{2,1}$ to store all level h binary intervals whose index contains $Lh - c\sqrt{LRh}$ or more 1s. We begin by counting the number of binary intervals whose index contains exactly $Lh - c\sqrt{LRh}$ 1s. If f, g are two functions of h we will say that f is asymptotically equivalent to g if $\lim_{h \rightarrow \infty} \frac{f(h)}{g(h)} = 1$. We will denote this relation by $f \sim g$. Standard estimates using Stirling's formula, which states that $k! \sim \sqrt{2\pi k} k^k e^{-k}$, lead to the proof of the following statement

$$B(h, Lh - c\sqrt{LRh}) \sim \frac{1}{\sqrt{2\pi LRh}} 2^{H(L)h} \left(\frac{L}{R}\right)^{c\sqrt{LRh}} e^{-c^2}$$

Next we compute the number of binary intervals whose index contains at least $Lh - c\sqrt{LRh}$ 1s.

LEMMA 1. *The number of level h binary intervals whose index contains at least $Lh - c\sqrt{LRh}$ 1s satisfies*

$$\begin{aligned} & \sum_{k \geq Lh - c\sqrt{LRh}} B(h, k) \\ & \sim \frac{1}{L-R} \sqrt{\frac{L}{2\pi Rh}} 2^{(H(L)h) \left(\frac{L}{R}\right)^{c\sqrt{LRh}}} e^{-c^2/2} \end{aligned}$$

Proof: It is easy to check that for any fixed k ,

$$\begin{aligned} & \frac{B(h, Lh - c\sqrt{LRh+k})}{B(h, Lh - c\sqrt{LRh})} \\ & \sim \frac{1}{(R/L)^k} \end{aligned}$$

hence the lemma follows from the sum for a geometric series with ratio R/L . *q.e.d*

Finally we compute the hit ratio one gets for a cache of the size appearing in the corollary. Let X_i be i.i.d random variables whose value is 1 with probability L and is zero otherwise. Since we have cached all the binary intervals which contain $Lh - c\sqrt{LRh}$ 1s or more and the probability of a zero or one in the index is distributed according to X_i , we need to calculate

$$P_h(c) = Pr(\sum_{i=1}^h X_i \geq Lh + c\sqrt{LRh})$$

According to the central limit theorem

$$\lim_{h \rightarrow \infty} P_h(c) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^c e^{x^2/2} dx = \Phi(c)$$

Putting it all together and expressing the results in terms relative to the size of the address space $n = 2^h$ we obtain the following result

THEOREM 1. *Fix a constant c . Consider a PQRS model for which $L = p + r > q + s = R$. As h tends to infinity, on a cache of size asymptotic to*

$$\frac{1}{L-R} \sqrt{\frac{L}{2\pi R \log(n)}} \left(\frac{L}{R}\right)^{c\sqrt{LR \log(n)}} n^{-(1-H(L))} e^{-c^2/2}$$

the algorithm $Alg_{2,1}$ will have in the limit a hit ratio of $\Phi(c)$.

The main term in the cache size is $n^{-(1-H(L))}$, hence on an order of magnitude level the hit ratio of the static greedy algorithm is controlled by the entropy of the spatial marginal a result which is expected.

remark 3: Since the number of addresses is 2^h practical values of h will certainly not exceed 80. for such small values it is actually much simpler and more accurate to compute all 40 binomial coefficients of the form $B(h, j) = B(h, h-j)$, $j = 0, \dots, 40$ directly and save them in a table. It is then very easy following the procedures outlined in the theorem to compute the hit ratios of the algorithm $Alg_{2,1}$ exactly for any given cache size rather than estimating them. This method would be recommended for all practical computations of hit ratios. The role of the estimates is to explain how hit ratios vary depending on the parameters and to provide a better conceptual understanding.

3.3 The optimal algorithm

The greedy static algorithms $Alg_{k,n}$ are efficient for self similar bias models, they are optimal static cache algorithms, however they are not optimal among all cache algorithms. For example, when $p = 1/8$, $q = 3/8$, $r = 1/4$ and $s = 1/4$ then the spatial marginal is uniform. For a uniform spatial distribution cache size and hit ratio coincide for any static cache algorithm. On the other hand if the bias model is not uniform, i.e. if one of the parameters p, q, r, s is not uniform,

then we can find cache algorithms which vary the content of cache with time which achieve asymptotic hit ratio 1 for any given fixed portion cache size.

recalling that a finite level bias model is an IRM at any given time point t we know that the optimal cache algorithm places in cache at time t the most likely addresses according to the IRM corresponding to t [2].

We now elaborate more explicitly which addresses will be cached at any given moment and what the expected hit ratio of this algorithm will be in the case of self similar binary bias models. Let $t = 0.t_0 t_1 \dots t_{h-1}$ and consider the binary time interval $I_t = [t, t + 1/2^h]$.

Let d equal 0 if $p > q$ and 1 otherwise, also let e equal 0 if $r > s$ and 1 otherwise.

Let

$$D_t = \{i \mid t_i = 0\}$$

and

$$E_t = \{i \mid t_i = 1\}$$

For a binary sequence m , let I_m denote the corresponding binary sub interval. For a pair of non negative integers (k, l) and some t we define

$$O_{k,l,t} = \cup_{m \in M_{k,l,t}} I_m$$

where $M_{k,l,t}$ is the set of all binary sequences m , of level h which when restricted to the positions in D_t differ from the constant sequence of value d in k places and when restricted to the complementary set of positions, E_t , differ from the constant sequence e in l places.

Let $c > 0$. We define the functional $F_c(k, l) = k + cl$ on the lattice of non negative pairs of integers (k, l) .

Consider the cache policy C_c which at time t orders the sets of addresses $O_{k,l,t}$ in descending order according to the value of $F_c(k, l)$ and places them in cache in that order until the cache is full.

THEOREM 2. *Let $c = \frac{|\log(p) - \log(r)|}{|\log(q) - \log(s)|}$ then C_c is optimal.*

Proof: At any given moment t the optimal policy places in cache the most likely binary level h intervals, [2]. The most likely interval is the one with value d on D_t and value e on E_t . Let us call the logarithm of it's probability v_t . If $m \in M_{k,l,t}$ then the logarithm of the probability of the interval I_m is

$v_t - k \log\left(\frac{Max(p,r)}{Min(p,r)}\right) - l \log\left(\frac{Max(q,s)}{Min(q,s)}\right)$. It is then easy to see that if $m_1 \in M_{k_1, l_1, t}$ and $m_2 \in M_{k_2, l_2, t}$ then $Pr(I_{m_1}) \geq Pr(I_{m_2})$ iff $F(k_1, l_1) \geq F(k_2, l_2)$. *q.e.d*

To tie the policies C_c to $Alg_{2,0}$ and $Alg_{2,1}$, we say that a PQRS model is consistent if $d = e$. If a PQRS model is consistent and $d = 0$ then $Alg_{2,0} = C_1$. If it is consistent and $d = 1$ then $Alg_{2,1} = C_1$.

We now compute the hit ratio of the optimal algorithm. recall that $n = 2^h$ is the size of the dataset. Unlike the case of $Alg_{2,0}$, $Alg_{2,1}$ we will content ourselves with computing a , the power of n , for which a cache of size $n^{a+\epsilon}$ has hit ratio close to 1, while a cache of size $n^{a-\epsilon}$ has hit ratio tending to 0 for all $\epsilon > 0$. Recall that H_S denotes the entropy of the spatial marginal while I denotes the mutual information of a model

THEOREM 3. *Consider a level h PQRS model, Then, for all $\epsilon > 0$ the hit ratio of the optimal cache algorithm tends to 0 when the size of the cache is $n^{-(1+I-H_S)-\epsilon}$ and tends to 1 when the size of the cache is $n^{-(1+I-H_S)+\epsilon}$.*

Sketch of proof: Almost all requests fall in time intervals whose binary expansion has approximately $(p+r)h$ zeros and $(q+s)h$ ones, so

$$|D_t| \sim (p+r)h$$

and

$$|E_t| \sim (q+s)h$$

The index of a typical spatial binary interval, restricted to D_t , will have $|D_t|p = (p+r)ph$ zeros and $|D_t|r = (p+r)rh$ ones. Similarly when restricted to the set of positions E_t it will have $(q+s)qh$ zeroes and $(q+s)sh$ ones. To contain a typical element, the logarithm of the size of the cache must therefore be asymptotic to

$$\log(B((p+r)h, (p+r)ph)B((q+s)h, (q+s)qh))$$

Which is asymptotic to

$$((H(p/(p+r))(p+r) + H(q/(q+s))(q+s))h = H_S - I$$

Normalizing the cache size by dividing by n we obtain the desired result. *q.e.d*

As an immediate corollary we have the following result

COROLLARY 1. *Among all bias models sharing a given spatial marginal distribution μ_S the I-models have the worst hit ratio with respect to an optimal caching algorithm.*

3.4 caching general bias models

So far we have introduced and analyzed cache algorithms for self similar bias models. We can extend the analysis to general bias models, the main differences being that there are no universal algorithms like *Alg* for general bias models, or even for random *PQRS* models. The description of the greedy static algorithm and the optimal algorithm also become less explicit.

As an example consider a binary bias model of level h whose spatial marginal has parameters $r_{i,l}$, $i = 0, 1$. Assume for simplicity that for all l , $r_{0,l} \geq r_{1,l}$ and define weights $w_l = \log(\frac{r_{0,l}}{r_{1,l}})$. Let $A = \{1, 2, \dots, h\}$. Order the subsets B of A by increasing order of the value of $f(B) = \sum_{i \in B} w_i$. The greedy static algorithm places in cache the data items whose addresses are given by the characteristic functions of the subsets B in the given order until the cache is full. It can be shown that the hit ratios are still controlled by the spatial entropy in the case of static algorithms and by $H_s - I$ in the case of an optimal algorithm.

3.5 singularity of time marginals and burstiness

The relation between burstiness of I/O requests and singularity properties of temporal marginals is equivalent to the relation between hit ratios and the singularity of spatial marginals. We may in fact define a model to be bursty if it's temporal marginal is singular. This means that almost all requests are given within short bursts which occupy together only a fraction of the total time of a trace. The preceding analysis of the spatial marginal can be carried over to the temporal marginal and calculates the relative portion of requests whose access times fall within the most heavily loaded time periods. In particular the entropy of the temporal marginal controls the over all burstiness, while $H_T - I$ controls the burstiness of a generic individual address.

4. SEEK DISTANCE CALCULATIONS

In this section we analyze seek distances in k-ary bias models. At first we assume that the bias model is used to model activity directed at the disk. We then analyze the seek distance in another setting which is more appropriate for read activity in which the output of the bias model first passes through a cache and only the misses reach the disk.

4.1 Seek distance calculations for bias models

We begin with a lemma which allows us to solve recursively for self similar models.

LEMMA 2. *Let F be a probability distribution on an interval $I = [0, a]$. Let $b \geq a$. Consider the shifted distribution $F_b(x) = F(x-b)$ on the interval $J = [b, a+b]$. Let S_F denote the average seek between a point in I and a point in J given by*

$$S_F = \frac{1}{a^2} \int_0^a \int_b^{b+a} (x-y) dF_b(x) dF(y)$$

then, $S_F = b$ for all F .

Proof: Consider the contribution of a pair of small intervals $[x, x+dx]$ and $[y, y+dy]$ and assume that in the definition of the integral we measure distance between the right endpoints, then, the contribution is $(x-y)(F_b(x+dx) - F_b(x))(F(y+dy) - F(y))$. If we consider the points $x-b \in [0, a]$ and $y+b \in [b, a+b]$ we get a contribution of $((y+b) - (x-b))(F(x-b+dx) - F(x-b))(F_a(y+b+dx) - F_a(y+b)) = (2b + (y-x))(F_b(x+dx) - F_b(x))(F(y+dy) - F(y))$. Adding the contributions we get the same contribution we would get from the constant function integrand b whose average value is obviously b itself. *q.e.d*

Let $E(S_P)$ denote the average seek distance between requests in a bias model with parameter set P .

THEOREM 4. *Consider a k-ary self similar bias model with parameters $p_{m,n}$ and $h = \infty$.*

1. Let

$$A = \sum_{m=0}^{k-1} \frac{\sum_{i,j=0}^{k-1} p_{m,i} p_{m,j} |i-j|}{\sum_{l=0}^{k-1} p_{m,l}}$$

and

$$B = \sum_{m=0}^{k-1} \frac{\sum_{i \neq j} p_{m,i} p_{m,j}}{\sum_{l=0}^{k-1} p_{m,l}}$$

$$\text{then, } E(S_P) = \frac{A}{k-1+B}.$$

In particular for the binary self similar bias model with parameters p, q, r, s . the average seek distance in the model $E(S_P)$ is given by $\frac{A}{1+A}$, where $A = 2(\frac{pq}{p+q} + \frac{rs}{r+s})$.

2. For a general k-ary bias model the average seek distance is given by

$$\sum_{l=0}^{\infty} \sum_{m=0}^{k-1} \sum_{i,j=0}^{k-1} \frac{p_{m,i,l} p_{m,j,l}}{\sum_{g=0}^{k-1} p_{m,g,l}} k^{-l-1}$$

The l 'th summand is at most k^{-l} .

3. The average seek distance at time t is given by

$$\sum_{l=1}^{\infty} \sum_{m=0}^{k-1} \sum_{j=0}^{k-1} \frac{p_{t_l,j,l} p_{t_l,j,l}}{\sum_{g=0}^{k-1} p_{t_l,g,l}}$$

4. Among all k-ary bias models which share fixed spatial and temporal marginal measures μ^T and μ^S the average seek distance is maximized by the I-model $\mu^T \times \mu^S$. The average seek distance of an I-model depends only on the spatial marginal μ^S and not on μ^T .

Proof: To simplify the exposition of the proof we prove part 1 for the case of binary bias models. The generalization to k-ary models is straightforward.

At any given time t we may compute the average seek between a request R_1 and the following (time wise) request R_2 inductively. Consider first a request which fell time wise in the interval $[0, 1/2]$. The next request time wise will also fall within this time interval with probability approaching 1 as the total number of requests tends to infinity. Consider the division of the spatial unit interval into the sub-intervals $I_1 = [0, 1/2]$ and $I_2 = [1/2, 1]$. The probability that the requests R_1 and R_2 fell into different sub-intervals is $1 - (\frac{p}{p+q})^2 + (\frac{q}{p+q})^2 = \frac{2pq}{(p+q)^2}$. Similarly the probability that two consecutive requests in the time interval $[1/2, 1]$ will fall into distinct spatial sub-intervals is $\frac{2rs}{(r+s)^2}$. The probability that two consecutive requests will fall into distinct spatial sub-intervals is then given by the weighted average

$$(p+q)\frac{2pq}{(p+q)^2} + (r+s)\frac{2rs}{(r+s)^2} \\ = 2\left(\frac{pq}{p+q} + \frac{rs}{r+s}\right) = A$$

At any given time t let F^t be the spatial distribution of points μ^t restricted to the interval $[0, 1/2]$ and normalized to a probability distribution. Let G^t be the normalized restriction of μ^t to the spatial sub-interval $[1/2, 1]$. We have $G^t = F^t_{1/2}$ in the notation of the lemma, that is the distributions within the sub-intervals are identical, this is easily verified from the construction of bias models.

By the lemma, applied with $a = b = 1/2$, the average seek time when restricted to requests in spatially distinct sub-intervals is $1/2$, hence the contribution of such pairs of requests to the seek average is $A/2$. The remaining pairs of requests R_1, R_2 whose probability is $1 - A$, fall into the same spatial sub-interval, either I_1 or I_2 . When the model is self similar then in both these sub-intervals the process averaged over all times is identical to that on $[0, 1]$ because of self similarity of the entire process, only the distances are halved. We get that the average seek $E(S)$ satisfies the equation $E(S) = (A/2) + \frac{1-A}{2}E(S)$ from which the result of part 1 follows immediately.

When the model is not self similar the same argument applies at each level l when A is replaced by $A_l = 2\left(\frac{p_l q_l}{p_l + q_l} + \frac{r_l s_l}{r_l + s_l}\right)$ and the distances are halved each time, yielding the series in part 2. The convergence rate statement is proved by noting that apart from the factor k^{-l-1} , the l 'th summand is a weighted average of the values $|i - j| \leq k$.

Similarly part 3 is given by the same argument applied to μ^t .

To prove part 4 consider the l 'th summand in the series expression for the average seek distance,

$$E(S)_l = \sum_{m=0}^{k-1} E(S)_{l,m} \\ = \sum_{m=0}^{k-1} \frac{\sum_{i,j=0}^{k-1} p_{m,i,l} p_{m,j,l} |i-j|}{\sum_{g=0}^{k-1} p_{m,g,l}}.$$

By [15] and [4] $E(S)_{m,l}$ is the total seek distance of an IRM with activity parameters $p_{m,i,l}$, $0 \leq i < k-1$. recall that the spatial marginal probabilities are given by

$$r_{i,l} = \sum_{m=0}^{k-1} p_{m,i,l} \\ \text{and that } \sum_{g=0}^{k-1} r_{g,l} = 1.$$

By theorem 1 of [4] the inequality

$$E(S)_l \leq \frac{\sum_{i,j=0}^{k-1} r_{i,l} r_{j,l} |i-j|}{\sum_{g=0}^{k-1} r_{g,l}}$$

$$= \sum_{i,j=0}^{k-1} r_{i,l} r_{j,l} |i-j|$$

will always hold if and only if there exist vectors

$V_i, 0 \leq i < k-1$ in \mathbf{R}^k such that

$$d(V_i, V_j) = \sqrt{|i-j|}$$

Here $d(V_i, V_j)$ denotes the standard distance between vectors. Let V_i be the vector whose j 'th coordinate is 1 for $j < i$ and 0 otherwise. It is easy to check that the V_i satisfy the desired condition. On the other hand consider an I-model with the same spatial marginal. The parameters of the I-model are given by

$$q_{m,i,l} = s_{m,l} r_{i,l}$$

where $s_{m,l}$ denote the marginal temporal probability parameters.

We have

$$\frac{q_{m,i,l}}{\sum_{g=0}^{k-1} q_{m,g,l}} \\ = \frac{r_{i,l}}{\sum_{g=0}^{k-1} r_{g,l}} \\ = r_{i,l}$$

hence,

$$E^I(S)_l \\ = \sum_m \left(\sum_g q_{m,g,l} \left(\sum_{i,j} \frac{q_{m,i,l}}{\sum_g q_{m,g,l}} \frac{q_{m,j,l}}{\sum_g q_{m,g,l}} |i-j| \right) \right) \\ = \left(\sum_{m,g} q_{m,g,l} \right) \left(\sum_{i,j} r_{i,l} r_{j,l} |i-j| \right) \\ = \sum_{i,j} r_{i,l} r_{j,l} |i-j|$$

The last expression shows that the average seek is independent of $s_{m,l}$ and dominates $E(S)_l$. Since $E(S)$ is a sum over all l of $E(S)_l$ we see that $E^I(S)$, the average seek of the I-model dominates $E(S)$ proving part 4. *q.e.d*

4.2 PQRS models after caching

We may think of the traces which a PQRS or bias model outputs as modeling an I/O request stream coming from a host computer. In most modern systems there will be one or more layers of cache between the host computer and the external disk drives on which data resides. When dealing with disk drive related quantities such as average seek distance we have to take into account the effect of the intermediate caches on the I/O stream. While write I/O for the most part will eventually be written to the disk, read I/O may be serviced by the intermediate caches and therefore read hits will not appear in the request stream as seen by a disk. Consequently, It is important to understand the Characteristics of the request stream of read misses from a cache when the input to the cache is a PQRS based trace. the resulting request stream of read misses will also depend on the cache algorithm which is employed. In this section we will analyze the output request stream of read misses from a cache which employs the static cache algorithm $ALG_{2,0}$. The analysis can also be carried out for more sophisticated cache algorithms such as the optimal cache algorithm but the case of $ALG_{2,0}$ is simpler and catches the main ingredients of the more general analysis. The main result is again stated in asymptotic form. The output will not be a PQRS based trace, however at any fixed level, the output will approach a PQRS based trace as the level h of the input increases. We can therefore say that the family PQRS algorithms is asymptotically closed under the operation of caching. If the original trace has parameters p_1, q_1, r_1, s_1 then the trace will approach a PQRS based trace with parameters p_2, q_2, r_2, s_2 which will depend on the original parameters and the size of the cache. For I-models as we increase cache size the seek distance will at first remain unchanged, then it will rise and after reaching a peak it will decrease until it reaches zero.

THEOREM 5. *Consider a PQRS model with level h approaching infinity and $p+r > q+s$. Consider the stream*

of read misses which is output by a cache of size $n^{H(v)-1}$ with algorithm $ALG_{2,0}$. If $v \geq p+r$ then the output stream of read misses will be asymptotically (as h tends to infinity) identical to the input stream. If $p+r > v$ then for any fixed level k the stream of read misses will asymptotically approach a PQRS model whose parameters p', q', r', s' are given by the equations $p/r = p'/r', q/s = q'/s'$ and

$$\frac{q' + s'}{p' + r'} = \left(\frac{q + s}{p + r}\right)^2 \left(\frac{v}{1-v}\right)$$

Proof: Let $L = p + r$ be the probability of having 0 in the i 'th spatial digit. By assumption $L > 1/2$. When $Alg_{2,0}$ is applied to a cache of size $n^{H(v)-1}$ it contains all addresses with at least $vh + 1$ zeroes. Read misses are the set of addresses with at most vh zeroes. Since almost all requests in the model have approximately vh zeroes the hit ratio will be nearly 0 if $v > L$, thus in this case as h tends to infinity the output sequence will be nearly identical to the input sequence as claimed. Assume $v < L$. Consider the set of misses whose most significant address bit is 0. This means that there are at most $vh - 1$ zeroes in the remaining $h - 1$ address bits. If the most significant address bit is 1, then a read miss will have at most vh zeroes in the remaining $h - 1$ bits. The total probability of the first set of misses (0 in the most significant bit) is

$$P_0 = L \sum_{j=0}^{vh-1} B(h-1, j) L^j (1-L)^{h-j-1}$$

the total probability for the second set of misses is

$$P_1 = (1-L) \sum_{j=0}^{vh} B(h-1, j) L^j (1-L)^{h-j-1}$$

Let

$$C(n, m, L) = \sum_{j=0}^m B(n, m) L^m (1-L)^{n-m}$$

and

$$A = \sum_{j=0}^{vh-1} C(h-1, j, L)$$

Using this notation, we have

$$P_0 = L \sum_{j=0}^{vh-1} C(h-1, j, L) = LA$$

and

$$P_1 = (1-L)A + (1-L)C(h-1, vh, L)$$

We consider the ratio of $C(h-1, vh, L)$ to $C(h-1, vh-1, L)$ as h tends to infinity.

$$\begin{aligned} & C(h-1, vh, L)/C(h-1, vh-1, L) \\ &= \left(\frac{1-v}{v}\right) \left(\frac{L}{1-L}\right) = (L/v) \left((1-v)/(1-L)\right) > 1 \end{aligned}$$

As h tends to infinity the ratio $C(h, vh-k, L)/C(h, vh-k-1, L)$ will converge to the same value for all fixed k , thus asymptotically $C(h, vh-k, L) = u^k C(h, vh, L)$, for $k = 1, 2, \dots$, where

$$u = \left(\frac{v}{L}\right) \left(\frac{1-L}{1-v}\right) < 1$$

We conclude that asymptotically $\frac{1-u}{u} A = C(h-1, vh, L)$. This leads to $P_1 = \frac{1-L}{uL} P_0$ which after expansion yields the formula in the theorem. Considering now the spatio-temporal behaviour we note that if a read miss begins with a most significant pair of $(0, 0)$ it will still produce a miss if we change the most significant pair to $(0, 1)$, where the first bit in the pair is the address bit while the second is the time stamp bit. We see that the ratio of p to r and q to s will remain the same, in fact, this is a feature of the stationary nature of the cache algorithm. We have thus established the claims of the theorem for the most significant pair of bits. Consider now spatio-temporal addresses which begin with a fixed sequence of pairs (i_l, j_l) , $l = 0, \dots, k-1$ and consider the probabilities for read misses of the k 'th significant pair (i_k, j_k) given the sequence above. We can apply the same type of computations as for the most sig-

nificant bit. We first consider the spatial marginal distribution for the k 'th bit. Let m denote the number of times that $i_l = 0$, $l < k$. The total probability for read misses with initial bit sequence (i_l, j_l) and k 'th spatial bit equal to 0 is $P_{0,m,k} = L \sum_{j=0}^{vh-m-1} C(h-k-1, j, L) = LA$, while the probability for a 1 in the k 'th spatial bit is similarly $P_{1,m,k} = (1-L)A + (1-L)C(h-k-1, vh-m, L)$. The asymptotic ratio of $P_{0,m,k}$ to $P_{1,m,k}$ is computed as before and since m and k are fixed it is easy to see that they play no role. We conclude that the asymptotic ratio of probabilities of $P_{0,m,k}$ to $P_{1,m,k}$ is independent of the history (i_l, j_l) and independent of k as well, thus asymptotically we obtain a PQRS model as well, with parameters as computed above. *q.e.d*

5. DISCUSSION OF IMPLICATIONS

So far we have discussed some analytic results concerning bias models. In this section we would like to examine the implications of our results to possible applications of the models.

5.1 Optimization

In trying to solve large scale optimization problems in storage systems it is often the case that repeated calculations using the same class of models but with varying parameters need to be done [3]. Having analytical formulas saves the need for time consuming multiple simulations. being able to analytically compute quantities of significant performance impact such as seek distances and hit ratios analytically allows the applicabilion of bias models in such contexts.

5.2 benchmarking

One possible potential application of PQRS models or binary bias models is benchmarking. Most benchmarks are synthetic since deploying real production software in testing is often problematic. Benchmarks may not be typical workloads, however, they carry a disproportionate economic weight since they are used for making large scale purchasing decisions. Since PQRS models look and feel "real" in the sense that they closely mimic bursty behavior, spatio-temporal dependencies and locality it would seem very tempting to use them as benchmarks. The results of this paper point to some precautions which must be taken with this approach. A self similar bias model has very few parameters. These parameters would reveal themselves very quickly once the trace is running. In fact if the basic time unit involved is known then once 2 out of the 2^h time units have passed, all the parameters can be estimated, and after a few more time units, they will be known with great certainty (assuming the I/O rate is reasonable). A clever storage vendor would be able to use this data to apply the optimal cache algorithms which are studied in the paper and to improve artificially the performance of his system. Even when the basic time unit is not known an analysis of auto correlations will reveal repeating patterns in a spectrum consisting of multiples of the basic time unit by powers of 2 and this information will quickly yield the basic time unit. Picking a more general bias system for modeling the workload allows more flexibility in terms of adjusting the entropy plots and will also reveal the parameters of the system more slowly. Even in this case almost all the relevant parameters will be known by the middle of the benchmark and even earlier. A possibly better solution to this pre-

dictability problem is to consider an even larger class of models for benchmarks which will allow the introduction of more randomness and unpredictability. As an example consider the following class of inhomogeneous PQRS models. The parameter set of an inhomogeneous PQRS model consists of the parameters p, q, r, s as before and functions $f_l, g_l, 0 \leq l < h$. f_l, g_l are functions from pairs of level l binary sequences $i = i_0, \dots, i_{l-1}, j = j_0, \dots, j_{l-1}$ with values in the set $\{0, 1\}$. We recall the 4 entropy preserving permutations, which we denote, $\pi(0, 0), \pi(0, 1), \pi(1, 0), \pi(1, 1)$ on the numbers p, q, r, s , which as recalled are identified with $p_{0,0}, p_{0,1}, p_{1,0}, p_{1,1}$. $\pi(0, 0)$ is the identity permutation. $\pi(1, 0)$ reorders p, q, r, s as q, p, s, r . $\pi(0, 1)$ reorders them as r, q, p, s and $\pi(1, 1)$ orders them as r, s, p, q . Given a level h sequence i denote it's level l prefix by i^l . We may define an inhomogeneous bias model by defining the probability of a level h sub square, $A_{i,j}$ to be

$$\prod_{l=0}^{h-1} \pi(f_l(i^l), g_l(i^l))(p_{i_l, j_l})$$

Thus the probabilities are permuted in each level in a manner which depends on the sequence indices i, j .

Since all the permutations are entropy preserving it is easy to check that the marginal entropies and mutual information of the process remain the same, hence the burstiness, spatio-temporal dependencies and locality properties remain similar. It is however much more difficult to predict specifically a good caching algorithm for such models especially when the functions f_l and g_l are pseudo random functions.

The average seek computations we presented do not hold for inhomogeneous models. They will depend specifically on the choices of f_l and g_l for small values of l . As the hit ratios depend on all levels l we may fix the functions f_l and g_l for small values of l to be constant functions thus retaining (to a large extent) both the average seek properties and the hit ratio properties of the original PQRS model.

6. BIAS IN REALITY

We have seen that bias models produced bursty I/O patterns with spatio-temporal dependencies and localized access, mimicking the behavior of many real I/O access patterns. The mechanism that causes these features is the bias in the digits of the addresses and access times of requests. We may wonder whether such a mechanism is also responsible for such phenomenon in real traces or whether the similarity to real traces is only at the level of the resulting features. A positive question would have some interesting consequences. If biases exist then they can be exploited. Caching algorithms can use this information to prefer higher probability addresses in a manner similar to the algorithms $Alg_{k,n}$ or the optimal dynamic algorithms.

In this section we consider the question of the existence of biases. We stress that the models can be useful in mimicking I/O patterns regardless of the answer to this question.

We decided to concentrate on binary biases since they seemed a bit more likely due to the special significance of powers of 2 in computer science. It is hard to imagine that biases in less significant binary digits of the access time (time stamp) will exist. The reason is that such biases depend among other things on the units in which time is measured and there is no natural atomic unit of time for I/O requests. Bias in the most significant digits will in general exist simply because the I/O rate varies over time, but beyond this important fact we do not expect other time biases to occur and we have found none.

We turn our attention to space addresses. They do contain a natural atomic unit, the block, and addresses are indeed given in blocks. In addition many applications will carry a basic page size. When the size of a page is $2^m k$ blocks, for k odd, the m least significant digits of the initial block of an I/O request will show maximal bias. If the pages are unaligned with the blocks or their size is an odd number of blocks then no bias should occur in the least significant digits of the initial block. In either case if we examine all blocks in the request rather than the starting block, the bias should be substantially less significant or non-existent. As with time the most significant digits of the address may be highly biased. This is an important phenomenon which is nearly pervasive in I/O workloads. I/O workloads are usually not uniformly distributed across the entire address range. In most real traces a certain region which is considerably smaller than the entire address space captures much of the total activity of the data set. The most significant digits of addresses in such a region will be fixed and create strong biases, which can indeed be exploited by caching algorithms. We thus have natural mechanisms for the existence of strong binary biases in the least and most significant digits of the address. The most exotic type of bias would hypothetically occur in the middle digits, say, bits 7-14 for a dataset measuring 1GB or more. Such biases would merit a special investigation since we do not know of a natural mechanism for creating them.

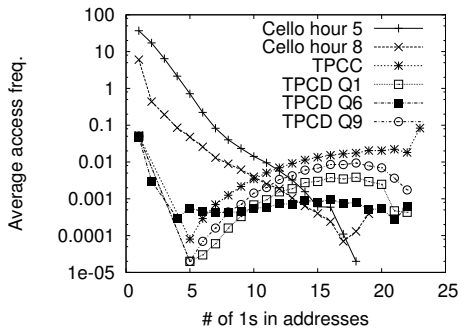
We present some of the evidence which we have collected regarding bias of addresses in real I/O workloads. We first examine the following two traces which were used in other studies.

The **Cello99** trace represent one year of user activity in 1999 on the main file server at HP Labs. This is a typical research group workload, including software development, trace analysis, and simulation. The trace has been used by various studies [7]. We study two (one hour) segments of traffic on a specific device 0x1f06800 from the trace, denoted as **Cello99 Busy** and **Cello99 Idle**. The specific start times of the trace segments are 4am and 7am, Feb. 1, 1999, respectively.

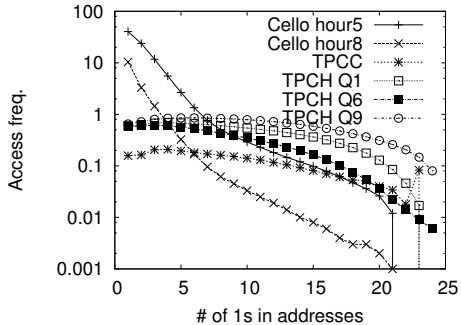
The other workload is collected from a TPC benchmark running on DB2 v.7.2. The database server is configured with 36 GB Maxtor Atlas 10K III disk running RedHat 7.3 distribution under Linux kernel v.2.4.19. This workload is used by [12]. The **TPCC** trace is a 10 minute run of the TPC benchmark. The **TPCH** trace contains isolated runs of 22 queries in the TPCH benchmark. We show results of three representative queries, **Q1**, **Q6**, and **Q9**.

Figure 1 shows average access frequencies of disk blocks grouped by the number of 1s in the addresses. the access frequency of a set of addresses is defined to be the number of I/O requests to the set divided by the size of the set. This normalization allows us to compare I/O traffic to subsets of the address space of different sizes. The graphs show the average access frequency, y , of requests whose addresses contain x 1s. (a) shows the bias for the starting block address of disk requests, and (b) for all the accessed blocks, i.e. all the contiguous blocks accessed by a disk request of certain size.

We observe that strong bias exists for starting address across all the workloads. When all the blocks of a request are considered, the bias stays strong for **cello99** traces, an observation which is rather surprising (and so far unex-



(a) Bias in starting addresses



(b) Bias in all accessed blocks

Figure 1: Bias in disk accesses by the number of 1s in the addresses.

plained). The database workload exhibits insignificant bias. The reason as we explained previously is that database systems organize data into Pages whose size is usually a power of 2. We observe almost uniform distribution in the TPCC trace since it is composed mostly of random accesses over the entire range of blocks. TPCD on the other hand involves large runs of sequential scans. Both these workloads thus result in uniform distributions of accesses grouped by the number of 1s in the addresses.

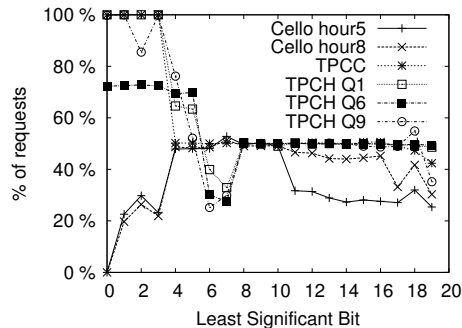
Figure 2 shows the bias of disk accesses by individual bits in the addresses. That is, the graphs show the percentage of accesses that have 1 on the x -th bit of the address starting from the least significant bit indexed by 0.

Similar to our previous analysis, we observe that strong bias exists for the starting addresses, especially at the left of the graphs. On the other hand, the all accessed blocks show no bias at all for low bits in the addresses.

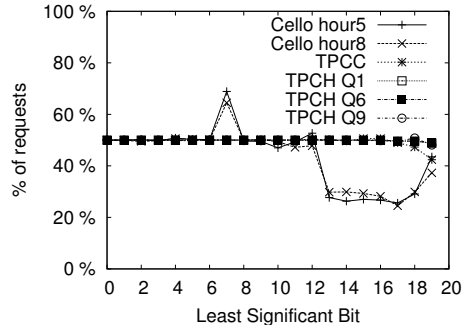
We note that our data contains an exotic bias in bit 7 (the 8th least significant bit) which merits further investigation.

Such events (and other level dependent biases) go unnoticed when only PQRS models are used. The derivative of the entropy function at the point $1/2$ vanishes, hence it is difficult to distinguish between no bias and, say, a 60-40 bias and the small effect on the entropy further diminishes when averaged over all levels. Bias models can capture such phenomenon more faithfully.

We also considered 107 randomly sampled production workloads from disk arrays, typically from large institutions and companies. The data in each disk array is subdivided into volumes whose size is typically a few GB. We considered the bias in the bits of each unit separately leading to 1103 separate traces on which we examined bias. Not all volumes



(a) Bias in starting addresses



(b) Bias in all accessed blocks

Figure 2: Bias in disk accesses by individual bits in addresses.

are independent from each other, so the effective volume sample size is actually somewhat smaller, still this is by far the most extensive survey conducted with production traces. We only considered traces which had more than 10,000 requests to eliminate random deviations from affecting our analysis. The traces come from, Unix, Windows NT and Mainframe environments and display a very wide range of behaviors. Apart from 3 distinctive traces we have no information about the applications running the traces, nor do we know the number and type of servers which generated the traces. The collection of traces is thus representative of production workloads in large installations as viewed by the disk array. Disk arrays typically have no semantic understanding of the data which resides on their disks and have no knowledge of applications which generate and manipulate the data. A full examination of the traces and their characteristics (request distribution, burstiness, hit ratios, alignment, sequentiality, request length, read/write ratios and others) will be presented elsewhere.

Of the 107 traces examined, 96 had no interesting bias patterns in any of their volumes, beyond the page size bias of least significant bits and locality bias of most significant bits. Sometimes the bias in the low and high bits is not complete indicating the existence of more than one active work area in the volume and more than one page size. This is expected since volumes may contain data belonging to different applications, or different types of data of the same application.

We conclude that the vast majority of workloads are not fundamentally biased, although circumstantial biases are prevalent.

In the remaining 11 traces which contained volumes with

interesting bias patterns not all volumes were necessarily biased. Only 44 of the 115 sufficiently active volumes in these arrays were biased in a significant way. A detailed analysis of the cause of bias in these 44 volumes is beyond the scope of the present paper. We restrict ourselves to a few remarks. Among the 44 biased volumes about half are lightly biased in the sense that while the bias is statistically significant in these volumes it is in the 40-60 percent range in the middle bits. The other half are strongly biased in nearly all bits. We include as an example the data relating to the bias of two volumes A and B, which belong to the same trace. Eight of the nine volumes with significant activity which belonged to this trace displayed interesting bias patterns, however the patterns differed between volumes.

The biases for bits 0-22 for volume A are
0.76, 0.75, 0.70, 0.70, 0.45, 0.49, 0.48, 0.53, 0.47, 0.52, 0.50, 0.59, 0.46, 0.35, 0.46, 0.47, 0.62, 0.54, 0.81, 0.66, 0.31, 0.81, 0.85

This is an example of a volume with mild bias in bits 4-12. The bias is significant in all bits since the bias range should be 0.49-0.51 in this case.

The biases for bits 0-22 for volume B in the same trace are

0.88, 0.88, 0.53, 0.38, 0.45, 0.55, 0.71, 0.48, 0.38, 0.38, 0.75, 0.70, 0.26, 0.65, 0.75, 0.26, 0.42, 0.22, 0.80, 0.40, 0.82, 0.81, 1.00

As may be observed volume B is heavily biased through all bits. Furthermore, the strength of the bias which is measured by the difference from the value 0.50 shows no definite trend. The fact that in both cases the bias is not strict in the low bits suggests that there are several workloads with different page sizes working concurrently in these volumes. Multiple localized workloads can be a source of consistent bias. We note however that there are other examples with a distinctive page size which are strongly biased

7. SUMMARY AND FUTURE WORK

In this paper we have computed analytically, rather than experimentally, seek distances and hit ratios for the recently introduced PQRS models of I/O access patterns. A new, more flexible, class of models, the bias models, were introduced and similarly analyzed. Our results may help further the use of such models in both optimization and benchmarking procedures. We examined what happens to traces produced by such models after they pass through a cache. We have also examined the question of whether fundamental biases really exist and saw that for the most part they don't.

We hope in the future to further explore both theoretical and practical aspects of bias models. In particular, we would like to analyze the entire seek distance distribution rather than its average. This will help us further in analyzing response times analytically. Towards the same goal we would like to analyze more closely the G/M/1 queueing model with arrivals given by a singular temporal marginal distribution. We can use the methods of the paper to compute maximal queue lengths but more refined information on the distribution of busy cycle periods would pose a bigger challenge.

One of the most interesting theoretical problems is to explore the relations between the algorithms $ALG_{k,n}$ for different k . We can show using the methods developed in [8] that they are largely independent.

8. REFERENCES

- [1] Aven O.I., Coffman E.G. and Kogan Y.A. *Stochastic analysis of computer storage*, D.Reidel publishing, 1987.
- [2] Aho A.V., Denning P.J. and Ullman J.D. Principles of optimal page replacement *J. of the ACM* 18, 80-93, 1971.
- [3] Anderson E., Hobbs M., Keeton K., Spence S., Uysal M. and Veitch A. Hippodrome: Running circles around storage administration *Proc. of the Conf. on file and storage Tech., FAST 02*, 175-188, 2002.
- [4] Bachmat E.. Recent results in mathematical modeling and performance evaluation of disks and disk arrays *Performance evaluation review* 28(4),, 24-26, 2001.
- [5] Borowsky E., Golding R., Jacobson P., Merchant A., Schreier L., Spasojevic M. and Wilkes J. Capacity planning with phased workloads in *1st workshop on software and performance (WOSP98)*, 199-207, Santa Fe, 1998.
- [6] Coffman E.G. and Denning P.J. *Operating systems theory*, Prentice Hall, 1973.
- [7] Gómez M. and Santonja V. A New Approach in the Modeling and Generation of Synthetic Disk Workload *MASCOTS'00*, 2000.
- [8] Host B. Nombres normaux, Entropy, *Translations Israel J. of mathematics* 91, 419-428, 1995.
- [9] Hutchinson J.E. Fractals and self similarity *Indiana U. Math. J.* 30, 713-747, 1981.
- [10] Lalley S.P Traveling salesman with a self similar itinerary *Probab. Engin. Infor. Sciences* 4, 1-18, 1990.
- [11] Opderbeck H. and Chu W.W. The renewal model for program behavior *SIAM J. of computing* 4, 356-374, 1975.
- [12] Schindler J., Ailamaki A. and Ganger G. Lachesis: Robust Database Storage Management Based on Device-specific Performance Characteristics *VLDB'03*, 2003.
- [13] Shannon C.E. and Weaver W. *Mathematical theory of communication*, U. of Illinois press, 1963.
- [14] Wang M., Ailamaki A. and Faloutsos C. Capturing the spatio-temporal behavior of real traffic data *Performance 2002*, Rome 2002.
- [15] Wong C.K. *Algorithmic Studies in mass storage systems*, computer science press, 1983.