Sensitivity Analysis of Frequency Counting*

Theodore Johnson
Dept. of Computer and Information Science
University of Florida

Abstract

Many database optimization activities, such as prefetching, data clustering and partitioning, and buffer allocation, depend on the detection of hot spots in access patterns. While a database designer can in some cases use special knowledge about the data and the users to predict hot spots, in general one must use information about past activity to predict future activity. However, algorithms that make use of hot spots pay little attention to the way in which hot spot information is gathered, or to the quality of this information. In this paper, we present a model for analyzing hot spot estimates based on frequency counting. We present a numerical method for estimating the quality of the data, and a rule-of-thumb. We find that if b of the references are made to the hottest a of the N data items, then one should process $Na\lceil(1-a)/(b-a)^2\rceil$ references.

1 Introduction

Many database optimization activities require the identification and classification of *hot spots*, or regions of the database with an exceptionally high access frequency. In some cases, a database designer can identify hot spots based on special knowledge of the data and the users. However, one is usually forced to predict future referencing patterns based on past behavior. A common approach is to use *frequency counting*. The number of references to the objects in the system are counted over a period of time. Objects with a high frequency count are labeled "hot", and the other objects are labeled "cold".

Frequency counting will accurately identify the hot objects in the database if the reference pattern is stationary and if enough references are collected. However, one usually wants to make database organization or re-organization decisions based on limited data. The earlier the hot spots are detected, the sooner the database can be optimized and the corresponding performance benefits obtained. In addition, reference patterns are usually not stationary, and decisions based on stale data can degrade performance. Since one is forced to make decisions based on limited data, a method of evaluating the quality of the hot spot information is needed.

In this paper, we examine the performance of frequency counting. The input to the frequency counting algorithm is a *reference string* composed of the sequence of object references that the system generates.

*We acknowledge the support of USRA grant #5555-19. Part of the work was performed while Theodore Johnson was an ASEE Summer Faculty Fellow at the NASA's National Space Science Data Center

We assume the Independent Reference Model (IRM), so that each reference is independent and identically distributed. Given the length of the reference string and the distribution of references, we calculate the quality of the hot spot estimate. One measure of the estimate quality is the probability that a hot object is labeled "hot". A second measure of quality is sum of the reference probabilities of the objects that are labeled "hot". Our objective is to calculate these measures, with an emphasis on the second. Towards this end, we derive procedures for calculating both performance metrics, and a rule of thumb for estimating when enough references have been processed.

1.1 Background

An accurate knowledge of the access frequency of objects in a database permits an optimal database organization. We present a brief survey here.

Suppose that references to database objects in a reference stream are independent and chosen from a stationary distribution. In this case, the optimal page replacement algorithm is the A_0 algorithm, which locks the most frequently accessed data objects into memory [6]. The layout of objects on a disk drive that minimizes head travel time is the "organ pipe arrangement" (place the most frequently accessed object in the middle, and more frequently accessed objects are placed closer to the center than less frequently accessed objects) [28].

Shared-nothing parallel database systems need to decluster the data stored in a relation, and scatter it across the processors so that the proper number of processors is involved in an average query. The declustering algorithms make use of statistics on average queries as part of their input [11, 7].

An object-oriented database typically consists of a number of objects which can have links to one another. Queries on the object base typically follow a path through the object links. As a result there has been much interest in placing objects that are 'close' on the same page, to reduce the page miss rate. Many researchers have observed that some links are more often traversed than others, and have proposed algorithms that use this kind of information [26, 17, 16, 25]. Tsangaris and Naughton [27] have found that stochastic clustering out-performs other methods. The reference stream for accessing blocks in a file system often shows a great deal of regularity. Several authors have proposed algorithms that observe patterns in the reference stream, and pre-fetch blocks that were often requested after the previous request [8, 19, 22, 21]. For prefetching and object clustering, one does not count object accesses, rather one counts pairs of references. Hence, the set of objects is correspondingly larger and the available data correspondingly scarcer.

In the above cited works, there is little analysis of the quality of the access frequence data. In this work we make a contribution to determining when one has collected enough access frequency data to make

clustering, declustering, migration, and prefetching decisions. We would like to make note of a related work by Salem, Barbara, and Lipton [23]. These authors consider the problem of identifying the most frequently accessed objects in a very large reference string while using a limited amount of work space. In contract to the work by Salem, Barbara, and Lipton, we assume that we already have accurate access frequency counts and we ask how well these counts reflect the distribution that generated them.

There is also a considerable body of related work in the Statistics literature. We defer a comparison between this work and related statistics work until Section 6.

2 Problem Statement

We assume that the database consists of N objects, labeled $d_1, d_2, \ldots, d_N$. We are given a reference string of length M, $R = (r_1, r_2, \ldots, r_M)$. The reference string is generated by the independent reference model [1], so that $\Pr[r_i = d_j] = p_j$ for all r_i in R. We denote the distribution $\{p_j\}$ by P, and we assume that $p_i \leq p_{i+1}$ for $i = 1 \ldots, M-1$.

We would like to identify the b "hottest" objects, d_1 through d_b, based on the information in R. Since we are restricted to using the information contained in R, we rank objects according to the number of references they received in R. For every d_i, we compute $ref(d_i)$ to be the number of references to d_i in R. For every $i = 1, \ldots, M$ we compute $pos(i)$ to be the d_j such that $ref(pos(i)) \geq ref(pos(i+1))$ (if two objects have the same reference count, we break ties arbitrarily). The b items with the most references are $pos(1)$ through $pos(b)$. Finally, we define the *rank* of a data item, $rank(d_i)$ to be the inverse mapping of pos, $pos(rank(d_i)) = d_i$.

After we process our reference string, we would like to determine how good our ranking of the data items is. There are several ways to measure a good ranking:

1. The quality of a ranking can be calculated by counting the number of data items d_1 through d_b which are ranked b or less (i.e, how many hot data items did we identify?). However, this measure does not capture well the meaning of a "good ranking" that we are after, since some hot items are fairly cool and some cool items are fairly hot. For example, suppose that the distribution is $\vec{p} = (.4, .3, .299, .001)$. Choosing (d_1, d_3) to be our hot items is judged to be a poor choice, but intuitively it is almost as good as (d_1, d_2). Also, (d_2, d_4) is measured to be as good as (d_1, d_3), but is actually much worse.

2. The quality of ranking can be calculated by summing the reference probabilities of $pos(1)$ through $pos(b)$. We call this measure the *buffer value*. It more directly reflects the intuitive idea of a good ranking. In the above example, (d_1, d_2) has a buffer value of .7, (d_1, d_3) has a buffer value of .699, and

(d_2, d_4) has a buffer value of .301.

We are mostly interested in buffer value as a function of M, but we also calculate the probability that a hot item is in the buffer, and the expected ranking of a data item.

3 The Model

Let us first try a direct approach. An object d_i is ranked s if receives more references than $N-s$ other objects and fewer references than $s-1$ others. Let $ref(j)$ be the number of references received by object d_j, with distribution F_j and density f_j. Let $\mathcal{P}(s-1,i)$ be the set of all partitions of $\{d_1, d_2, \ldots, d_{i-1}, d_{i+1}, \ldots, d_N\}$ into $A \cup B$ such that $|A| = s-1$. Let $\mathcal{R}(s,i)$ be the probability that object d_i is ranked s. Then [9]

$$\mathcal{R}(s,i) = \int_0^\infty \sum_{\mathcal{P}(s-1,i)} \left[\prod_{j \in A} (1 - F_j(x)) \prod_{j \in B} F_j(x) \right] f_i(x) dx$$

This integral is extremely difficult to evaluate for a large or an arbitrary number of objects in the system. In addition, the distribution functions F_j do not have closed form for their exact value. Therefore, we are compelled to search for approximate answers.

3.1 An Approximate Approach

Let us first look at the the distribution functions. Since object d_j is accessed independently on every reference with probability p_j, the density function of the number of accesses over M references is binomial(M,p). The distribution is therefore an incomplete binomial sum, and does not have a closed form [13]. We instead use the normal approximation to the binomial distribution:

$$\text{binomial}(M,p) \approx N(Mp, Mp(1-p))$$

The factor of $(1-p)$ in the variance of the normal distribution creates difficulties in the subsequent calculations. The value of p is typically very small – a data item with $p = .1$ is extremely hot. Therefore we make the further approximation that binomial$(M,p) \approx N(Mp, Mp)$

Two objects d_i and d_j do not receive independent numbers of references. However, this dependence generally has a very small effect, and is cumbersome to account for. We make the approximation that all objects receive references independently of the number of references to other objects.

To compute the ranking of a data item d_i, we compare $ref(d_i)$ to $ref(d_j)$ for each d_j, and count the number of times that $ref(d_i)$ is larger (breaking ties arbitrarily). If we are told that $ref(d_i) > ref(d_j)$, then we know that d_i is likely to have a large reference count and thus a low ranking. So, the indicator variables

$I_{ref(d_i)>ref(d_j)}$ and $I_{ref(d_i)>ref(d_k)}$ are not independent, and in fact show a very strong dependence. If d_i receives K references, $rank(d_i)$ is is likely to be located in a small interval centered on a point that depends on K, $E[K]$.

We therefore take the following approach to testing the sensitivity of frequency counting. We ask, if data item d_i receives K references, what is the chance that it is ranked in the top b in terms of reference counts? We denote this probability by $\mathcal{R}_<(b|K)$. To determine the probability $\mathcal{R}_<(b,i)$ that object d_i is in the buffer, we uncondition on K:

$$\mathcal{R}_<(b,i) = \sum_{K=0}^{\infty} \mathcal{R}_<(b|K) f_i(K)$$

The Derivations

We first list the important symbols that we use in the analysis:

- N : Number of data items.
- d_i : Data item i, $1 \leq i \leq N$.
- R : Reference string.
- $M = |R|$.
- b : Size of the buffer (number of items to mark as hot).
- $ref(d_i)$: Number of references to d_i in R.
- $pos(j)$: The index of the data item that is ranked j.
- $rank(i)$: The ranking of d_i.
- F_i (f_i) : The distribution (density) of $ref(i)$.
- p_i : Probability that a reference is to item i.
- *buffer value* : $\sum_{j=1}^{b} p_{posj}$.
- $\mathcal{R}(s,i)$: The probability that d_i is ranked s.
- $\mathcal{R}_<(b,i) = \sum_{s=1}^{b} \mathcal{R}(s,i)$.
- $\mathcal{R}_<(b,K) = \mathcal{R}_<(b,i|d_i = K)$.
- $E[K]$: Average ranking of an object with k references.

- $V[K]$: Variance of the ranking of an object with K references.
- $V^I[K]$: Variance in $E[K]$.
- $V^T[K]$: Variance due to tie breaking.
- $p_{(-1)}, p_{(+1)}$: Lower and upper bounds of integration wrt. p_j.
- j_{lo}, j_{hi} : Lower and upper bounds of integration wrt. j.

To calculate $\mathcal{R}_<(b|K)$, we take the following approach. For every object d_j, we calculate the probability that the object will have a lower rank than an object with K references as $X_j[K] = \Pr[ref(j) > K] + \Pr[ref(j) = K]/2$ (recall that $ref(j)$ is the random variable corresponding to the number of references to d_j, and we break ties arbitrarily when ranking). Each object d_j makes its own contribution to the number of objects with more than K references. The total number of objects ranked lower than an object with K reference has mean $E[K] = \sum X_j[K]$. If d_j receives K references, its ranking is a random variable with mean $E[K]$ and variance $V[K]$. We approximate the distribution of the ranking by a Normal distribution, so

$$\mathcal{R}_<(b|K) = \Pr[N(E[K], V[K]) \leq b]$$

We need to calculate $E[K]$. We have approximated the distribution of $ref(j)$ as a normal $N(Mp_j, Mp_j)$ distribution, which automatically captures the tiebreaking:

$$\begin{aligned} Pr[ref(j) > K] + Pr[ref(j) = K]/2 = &\quad Pr[N(Mp_j, Mp_j) > K] \\ = &\quad Pr\left[N(0,1) < \frac{Mp_j - K}{\sqrt{Mp_j}}\right] \end{aligned}$$

Even the Normal distribution is too difficult to work with. Fortunately there is a simple first-order approximation. We will approximate $N(0,1;x)$ by a linear approximation $N^*(0,1;x)$, which is non-constant only in $[-1,1]$. As a result, we find that:

$$Pr[ref(j) > K] \approx \begin{cases} 0 & \frac{Mp_j - K}{\sqrt{Mp_j}} < -1 \\ C1 + C2\frac{Mp_j - K}{\sqrt{Mp_j}} & -1 \leq \frac{Mp_j - K}{\sqrt{Mp_j}} \leq 1 \\ 1 & 1 < \frac{Mp_j - K}{\sqrt{Mp_j}} \end{cases} \quad (1)$$

where $C1 = .5$ and $C2 = 1/(\sqrt{2\pi}) \approx .3989$. We note that more accurate higher-order approximations can be used, at the cost of a more complex model.

We define $I_j[K]$ to be 1 if $rank(j) > K$ and 0 otherwise. Let $E_j[K]$ and $V_j^I[K]$ be the mean and variance of $I_j[K]$. Let $I[K] = \sum_j I_j[K]$. Then $I[K]$ has mean $E[K] = \sum_j E_j[K]$ and variance $V^I[K] = \sum_j V_j[K]$.

The sums required to compute $E[K]$ and $V^I[K]$ are usually intractable, so we approximate them by an integral over p_j. So, the first thing to determine are the interesting bounds of the integration. We note that $(Mp_j - K)/\sqrt{Mp_j}$ is an increasing function of p_j. By solving

$$\frac{Mp_j - K}{\sqrt{Mp_j}} = \pm 1 \tag{2}$$

we find that

$$\begin{aligned} p_{(-1)} &= \frac{1 + 2K - \sqrt{1+4K}}{2M} \\ p_{(+1)} &= \frac{1 + 2K + \sqrt{1+4K}}{2M} \end{aligned} \tag{3}$$

We need to convert $p_{(-1)}$ and $p_{(+1)}$ into bounds over d_j i.e., we need to find the $j_{(-1)}$ and $j_{(+1)}$ such that $p_{j_{(-1)}} = p_{(-1)}$ and $p_{j_{(+1)}} = p_{(+1)}$. As an example, let us specialize to a triangular distribution, $p_j = a + bj$ ($b < 0$). Note that since $b < 0$ (i.e, we order the hottest data items first), $j_{(-1)}$ is the index of the first d_j that is not 'assured' of receiving at least K references, and $j_{(+1)}$ is the index of the first d_j that is 'assured' of receiving less than k references. Then

$$\begin{aligned} j_{(-1)} &= \frac{1 + 2K - 2Ma - \sqrt{1+4K}}{2Mb} \\ j_{(+1)} &= \frac{1 + 2K - 2Ma + \sqrt{1+4K}}{2Mb} \end{aligned}$$

Computing $E[K]$ requires an integral of $E_j[K]$ over j. We need to define j_{lo} and j_{hi} as $j_{lo} = max(j_{(+1)}, 0)$ and $j_{hi} = min(j_{(-1)}, N)$. Then, the integral is

$$\begin{aligned} E[K] = & \int_0^{j_{lo}} 1 dj + \int_{j_{lo}}^{j_{hi}} C1 + C2\frac{Mp_j - K}{\sqrt{Mp_j}} dj + \int_{j_{hi}}^{N} 0 dj \\ = & \quad (j_{hi} + j_{lo})/2 + C2 \int_{j_{lo}}^{j_{hi}} \frac{Mp_j - K}{\sqrt{Mp_j}} dj \end{aligned} \tag{4}$$

The bounds of integration can fall into several ranges, as illustrated in Figure 1, each of which provide different answers. If $j_{lo} > N$ (case 6) then $E[K] = N$. If $j_{hi} = 0$ (case 1), then $E[K]$=0. The principle non-trivial cases are $0 < j_{lo} < j_{hi} < N$ (case 3) and $0 = j_{lo} < j_{hi} < N$ (case 2). Case 4 is degenerate, and we approximate case 5 by case 3.

Next, we turn to estimating the variance $V[K]$. There are two components to the variance in the ranking of a data items that receives K references. The first component is the variance in $E[K]$, which we write as $V^I[K]$. The second component of the variance is due to the random tiebreaking, which we write as $V^T[K]$. We calculate $V[K] = V^I[K] + V^T[K]$.

Since variances sum, we can compute $V^I[K]$ by summing the individual contributions of each $V_j^I[K]$,

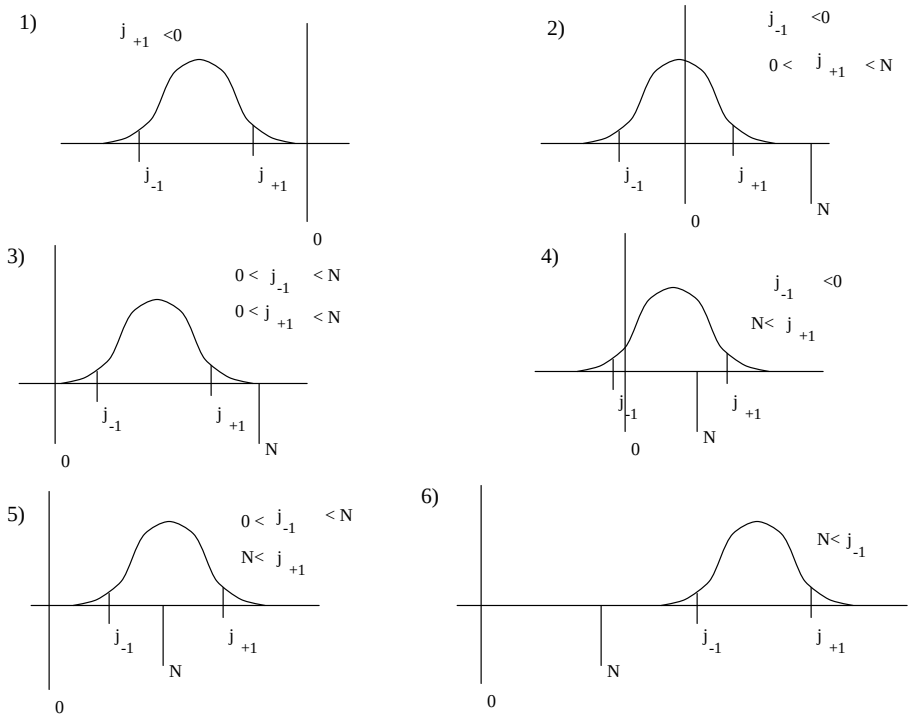

Figure 1: *Possibilities for the bounds of integration.*

which is $I_j[K]^2$. Thus,

$$V^I[K] = \frac{j_{hi}-j_{lo}}{4} - \int_{j_{lo}}^{j_{hi}} C2^2 \left(\frac{Mp_j - K}{\sqrt{Mp_j}}\right)^2 dj \tag{5}$$

If $Y[K]$ data items receive K references, then the ranking of a data item that receives K references is uniformly distributed in $[E[K] - Y[K]/2, E[K] + Y[K] + 2]$. So, the variance due to tiebreaking is $V^T[K] = Y[K]^2/12$. For small K, one can estimate $Y[K]$ by integrating $\Pr[C_j = K]$ over j. For large K, the binomial distribution or its Poisson approximation become cumbersome. In that case, once can approximate $Y[K] \approx (E[K] - E[K-1])/2 + (E[K+1] - E[K])/2 = (E[K+1] - E[K-1])/2$. $V^T[K]$ usually dominates $V^I[K]$.

We next calculate $E[K]$ and $V[K]$ for three distributions: triangular, exponential, and partitioned. The triangular density function has a gradual slope, while the exponential density function has a steep slope. The partitioned density function has a very steep, and is useful for approximating an unknown distribution.

3.1.1 Triangular Distribution

In a triangular distribution, $p_i = a + b * i$. We start by calculating the bounds of integration for $E[K]$ and $V^I[K]$:

$$j_{(-1)} = \frac{1 + 2K - 2Ma - \sqrt{1 + 4K}}{2Mb}$$
$$j_{(+1)} = \frac{1 + 2K - 2Ma + \sqrt{1 + 4K}}{2Mb}$$

The term $\sqrt{1 + 4K}$ makes the integral difficult, so we approximate the bounds by:

$$j^*_{(-1)} = \frac{1 + 2K - 2Ma + 2\sqrt{K} + 1/4\sqrt{K}}{2Mb}$$
$$j^*_{(+1)} = \frac{1 + 2K - 2Ma - 2\sqrt{K} - 1/4\sqrt{K}}{2Mb}$$

Where

$$j^*_{(-1)} = j_{(-1)} + O\left(\frac{1}{MbK^{3/2}}\right)$$
$$j^*_{(+1)} = j_{(+1)} + O\left(\frac{1}{MbK^{3/2}}\right)$$

When we substitute $p_i = a + b * i$ into equation 4, we get

$$E[K] = \quad (j_{hi} + j_{lo})/2 + C2\left[\frac{2(a+bj)^{1/2}(-3K+Ma+Mbj)}{3M^{1/2}b}\right]_{j_{lo}}^{j_{hi}}$$

When $j_{lo} = j_{(-1)}$ and $j_{hi} = j_{(+1)}$, we get

$$E[K] = \quad -\frac{-2\,K-1+2\,Ma}{2\,Mb} - \frac{2*C_2}{3MB}$$
$$= \quad \frac{K+1/2-Ma-2C_2/3}{Mb}$$

We note that determining $E[K]$ is not as simple as calculating the number of data items d_j such that $Mp_j > K$, since approach incorrectly calculates $(K - Ma)/Mb$ for $E[K]$.

When $j_{lo} = j_{(-1)}$ and $j_{hi} = j_{(+1)}$, we get

$$E[K] = \quad \frac{2\,K+1-2\sqrt{K}-2\,Ma}{4\,Mb} + -\frac{C2\,\sqrt{2}\sqrt{2\,K+1-2\sqrt{K}}\left(4\,K-1+2\sqrt{K}\right)}{6\,Mb} - \frac{2\,C2\,\sqrt{a}(Ma-3\,K)}{3\,b\sqrt{M}}$$
$$= \quad \left(\frac{2\,C2\,\sqrt{a}}{b\sqrt{M}} + \frac{1}{2\,Mb}\right)K - \frac{4\,C2\,K^{3/2}}{3\,Mb} + \frac{C2\,\sqrt{K}}{2\,Mb} - \frac{C2}{3\,Mb} - \frac{2\,C2\,a^{3/2}\sqrt{M}}{3\,b} + \frac{1-2\sqrt{K}-2\,Ma}{4\,Mb}$$

We next need to calculate $V^I[K]$. When $j_{lo} = j_{(-1)}$ and $j_{hi} = j_{(+1)}$, we get

$$V[K] = \quad -\frac{\sqrt{K}\left(1+4\,C2^2K-2\,C2^2-2\,C2^2K^{3/2}\ln(2\,K+1+2\sqrt{K})+2\,C2^2K^{3/2}\ln(2\,K+1-2\sqrt{K})\right)}{2\,Mb}$$

We want to make an approximation to remove the logs. We observe that

$$\ln\left(1 + \frac{\sqrt{4K}}{2K + 1 - 2\sqrt{K}}\right) = \quad \frac{2}{K^{1/2}} - \frac{1}{3K^{3/2}} + O\left(\frac{1}{K^{5/3}}\right)$$

We empirically find that the two-term approximation is very close to the exact value even for small K.

Putting this approximation into the formula for $V[K]$ gives us

$$\begin{aligned} V[K] = & -\frac{\sqrt{K}\left(1+4\,C2^2K-2\,C2^2-2\,C2^2K^{3/2}\left(\frac{2}{\sqrt{K}}-\frac{1}{3K^{3/2}}\right)\right)}{2\,Mb} \\ = & \frac{K^{1/2}}{Mb}\left(\frac{4C2^2}{3}-\frac{1}{2}\right) \end{aligned}$$

When $j_{lo} = j_{(-1)}$ and $j_{hi} = j_{(+1)}$, we get

$$\begin{aligned} V^I[K] = & \frac{-2Ma+C2^2-4C2^2a^2M^2+1-2\sqrt{K}+2K-4C2^2\sqrt{K}-12C2^2K^2+8C2^2K^2\ln((1+2K-2\sqrt{K})/(2Ma))+16C2^2aMK-8C2^2K^{3/2}}{8\,Mb} \\ \approx & \frac{-2Ma+C2^2-4C2^2a^2M^2+1-2\sqrt{K}+2K-4C2^2\sqrt{K}-12C2^2K^2+8C2^2K^2(\ln(K/(Ma))-K^{-1/2})+16C2^2aMK-8C2^2K^{3/2}}{8\,Mb} \end{aligned}$$

3.1.2 Exponential Distribution

In the exponential distribution, $p_j = c * r^j$, where $c = (r-1)/(r^N-1)$. After substituting for p_j into equation 3 and solving for j, we find that

$$\begin{aligned} j_{(-1)} = & -\frac{\ln(2)-\ln(1+2\,K+\sqrt{1+4\,K})-\ln(r^{N+1}-1)+\ln(M)+\ln(r-1)}{\ln(r)} \\ j_{(+1)} = & -\frac{\ln(2)-\ln(1+2\,K-\sqrt{1+4\,K})-\ln(r^{N+1}-1)+\ln(M)+\ln(r-1)}{\ln(r)} \end{aligned}$$

After solving equation 4, we find that

$$E[K] = (j_{lo}+j_{hi})/2 + \left.\frac{2\,C2\left(-r^{-\frac{j}{2}+N+1}K+r^{-\frac{j}{2}}K-r^{\frac{j}{2}+1}M+r^{\frac{j}{2}}M\right)}{\sqrt{M}\sqrt{r-1}\sqrt{r^{N+1}-1}\ln(r)}\right|_{j_{lo}}^{j_{hi}}$$

After solving equation 5, we find that

$$V^I[K] \approx \frac{C2^2\left((r^2-2r+1)M^2r^j+2\,jKr\ln(r)M-2\,jK\ln(r)M+2\,r^{-j+N+1}K^2-r^{-j}K^2\right)}{\ln(r)M(r-1)(r^{N+1}-1)}$$

3.1.3 Partitioned Distribution

In a partitioned distribution, the data items are partitioned into n parts. Partition i contains a_iN data items, and receives b_i of the references. Every data item in partition i is equally likely to be referenced.

Because the partitioned distribution has an inherently discrete description, we take a different approach to calculating $E[K]$ and $V[K]$.

1. set $[-1] = hits[-1] = hits[*] = 0$.
2. For every K,
   (a) For every partition i,

i. Calculate the expected number of data items in partition i that will receive K references using the Poisson approximation to the binomial distribution **Citation?**.

ii. Add this value to $hits[K]$.

(b) Set $E[K] = E[K-1] + (hits[K-1] + hits[K])/2$.

(c) set $V[K] = hits[K]^2/12$.

3.2 Performing The Computation

To calculate the performance measures of interest (the probability that a data item is in the buffer, and the probability value in the buffer), we use the following procedure:

1. Determine the range of K for which we need to compute $E[K]$ and $V[K]$. Set $Kmin$ to the K that satisfies $j_{(+1)}(K) = N$, and set $Kmax$ to the K that satisfies $j_{(-1)}(K) = 0$. If $Kmin < 0$, set $Kmin = 0$, and if $Kmax > M$, set $Kmax = M$.

2. For each K in $[Kmin \ldots Kmax]$, compute $E[K]$ and $V[K]$.

3. For each K, compute $\mathcal{R}_<(b|K) = \Pr[N(E[K], V[K]) \leq b]$

4. For each data item j, compute the probability that it is admitted to the buffer.

   (a) Compute bounds on number of references that d_j is likely to receive, $[Klo, \ldots, Khi]$. These can be estimated by the normal approximation to the binomial distribution.

   (b) Set $p_{admitted}[j] = 0$.

   (c) For each K in $[Klo, \ldots Khi]$,

      i. Compute the probability that d_j receives K references, $prob_i$, by using the Poisson approximation to the Binomial distribution.

      ii. Set $p_{admitted}[j] = p_{admitted}[j] + prob_i * \mathcal{R}_<(b|K)$

5. Compute a normalizing constant $normalize$ by summing $p_{admitted}[j]$ over j and dividing by B. Divide each $p_{admitted}[j]$ by $normalize$ (to reduce errors).

6. Compute $buffer_{value}$ by summing $p_j * p_{admitted}[j]$ across all j.

This procedure requires that we perform $O(1)$ operations for each K, and $O(max_j(Khi - Klo))$ operations for each j. We know that $max_j(Khi - Klo)$ is $O(\sqrt{Kmax})$, so the complexity of the calculation is $O(N\sqrt{Kmax}) = O(N\sqrt{p_1 M})$.

4 Results

To validate our model, we wrote a simulator that generated a reference string and ranked the data items according to the number of references they received, breaking ties arbitrarily. Based on the ranking, we calculated the probability that specific data items were ranked B or less, the reference probability of the data items ranked B or less, and the average ranking of a data item that received K references. We ran the simulator for 10,000 reference strings to compute expected values. We used a database of $N = 200$ items, set $B = 40$, and varied the length of the reference string.

We used three different probability distributions: triangular, exponential, and partitioned. The results for the triangular distribution are shown in Figure 2. In the triangular distribution, the probability of accessing data item i is $a * i + b$, and a and b have been assigned so that $p_1 = 10 * p_{200}$. In Figure 2, *min* is the value the buffer would have if items were placed in the buffer randomly, and *max* is the best possible buffer value. The agreement between simulation and analytical predictions is very close. The results for the exponential distribution are shown in Figure 3. Here, $p_i = c * r^i$, and $r = .97$. For small numbers of references, the analytical model underestimates the value of the buffer. However, for moderate to large numbers of references, the predictions are accurate. Figure 4 shows the results for the partitioned distribution. In this distribution, 80% of the references are made to 20% of the data items. The agreement between simulation and analytical predictions is good throughout the range of M. In general, the analytical model makes accurate predictions, though accuracy suffers when the reference string is small and the distribution p_i has a steep slope. The inaccuracies are due to the approximations made in the calculations.

We also measure the quality of a ranking by the probability that a hot item is included in the buffer. We ran experiments with the triangular, exponential, and partitioned distributions, and plot the probability that the 30^{th} hottest item is included in the buffer against length of the reference string. The comparison between analytical and simulation predictions is shown in Figure 5. The agreement between the simulation and analytic predictions is good.

5 A Rule Of Thumb

While the analytical model that we have presented is far faster than a simulation, a simple model that can be immediately applied and does not depend on the distribution $\vec{p}$ is more useful in practice. In this section, we present a simple rule of thumb for determining when enough reference have been collected.

Suppose that we want to buffer the aN hottest items, which receive b of the total references. We can look at a partitioned distribution where $p_{hot} = b/aN$ and $p_{cold} = (1-b)/(1-a)N$.

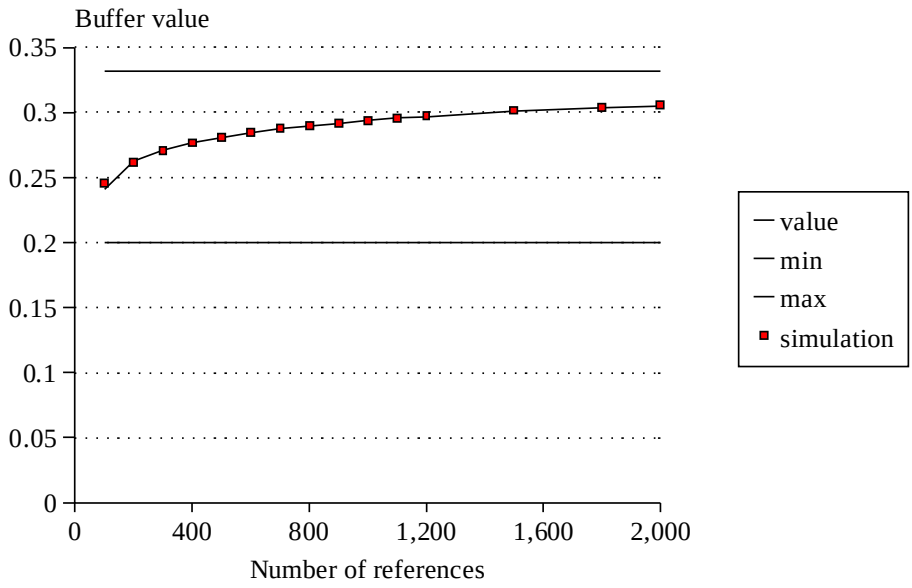

Figure 2: Comparison of buffer value predictions for a triangular distribution.

By solving equation 2 for K, we can determine the highest and lowest K for which the approximation 1 is non-constant. We denote these as K_{hi} and K_{lo}, and we find that

$$K_{hi} = Mp + \sqrt{Mp} \quad K_{lo} = Mp - \sqrt{Mp}$$

The value of M for which $K_{hi}(p_{cold}) = K_{lo}(p_{hot})$ is enough references to distinguish a hot item from a cold item in most cases. So, to find M_{cut}, we solve $K_{hi}(p_{cold}) = K_{lo}(p_{hot})$ for M to find

$$M_{cut} = \frac{Na(1-a)}{(b-a)^2} \tag{6}$$

We notice that Na is the size of the buffer. Therefore, formula 6 asks that we process X references for every item in the buffer, where

$$X = \frac{1-a}{(b-a)^2}$$

If X is usually non-integral. We can improve our estimate of the number of references we need to collect by processing $\lceil X \rceil$ references for every hot item. This leads us to:

Rule of Thumb: If the buffer holds aN items, and b of the references are directed to the aN items, then collect M_{ROT} references, where

$$M_{ROT} = Na \left\lceil \frac{1-a}{(b-a)^2} \right\rceil \tag{7}$$

Table 1 summarizes the predictions that the Rule of Thumb makes on the distributions in our validation experiments. The parameters a and b are the minimum and maximum possible buffer values. We applied

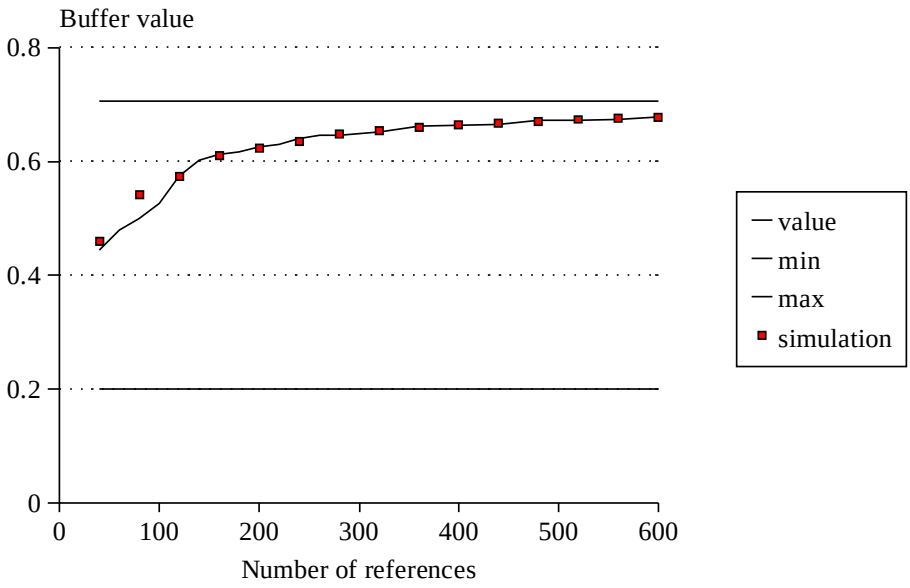

Figure 3: Comparison of buffer value predictions for a exponential distribution.

distribution	a	b	M_{ROT}	buffer value	percent
triangular	.2	.332	1840	.304	78.9
exponential	.2	.706	160	.613	81.6
partitioned	.2	.8	120	.623	70.5

Table 1: Summary of the Rule of Thumb predictions.

the Rule of Thumb to obtain M_{ROT}, and then reported the buffer value that our analytical model predicts. Finally, we report the quality of the ranking as the distance of the buffer value between the minimum and maximum, in percent. The table shows that the Rule of Thumb makes a reasonable prediction of the number of references to process, since the the buffer collects 70-80% of the additional buffer value that can be obtained by processing references. The table shows the one must collect more references to distinguish between hot and cold data items as the difference in reference probability becomes smaller. Thus, fewer than N references need to be processed for the exponential and partitioned distributions, while a huge number of references must be processed for the triangular distribution.

6 Comparison to Statistics Research

Statistics researchers working in the area of design of experiments have examined problems related to the one examined in this paper. A good survey of the area is contained in the book by Gibbons, Olken, and Sobel [12].

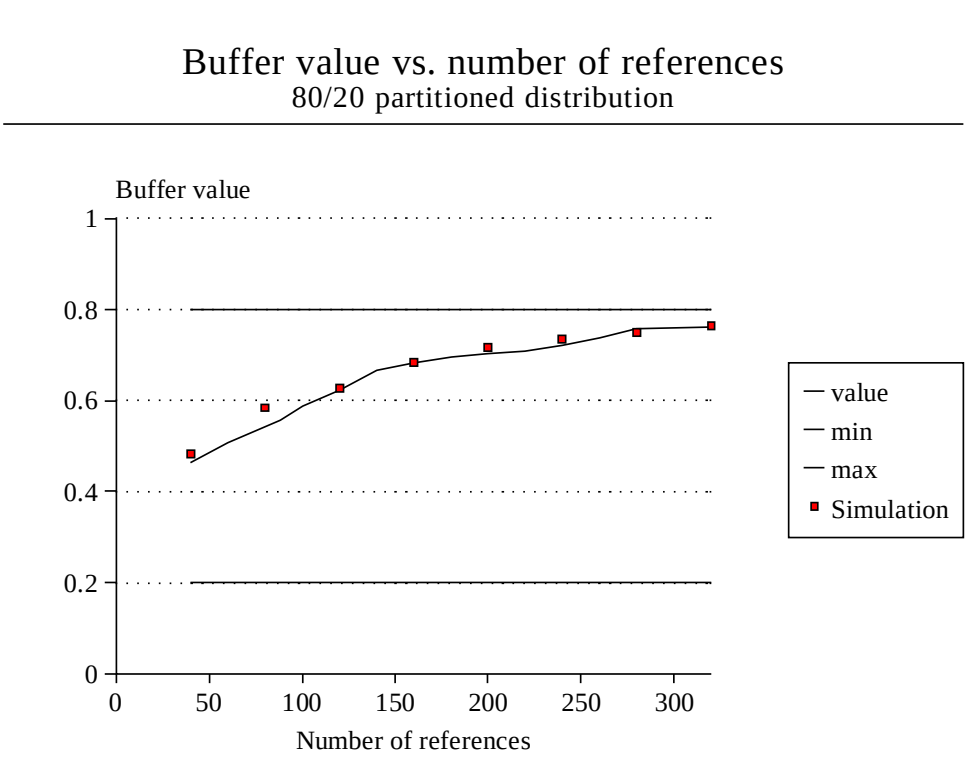

Figure 4: Comparison of buffer value predictions for a exponential distribution.

Bechhofer [2] worked on the original problems in this area. Given k populations, each with (unknown) mean X_i and (known) variance σ_i^2, find the t populations with the largest (best) means. The best populations can be ranked or unranked. Bechhoffer's procedure is to draw N_i samples from population i, $i = 1 \ldots k$, compute the sample means $\bar{X}_i$, and rank the sample populations accordingly. The problem becomes, how many samples N_i should I draw from each population to be certain that the t populations that I have selected are the t best with probability P? For the case when all variances are equal and the populations have a Normal distribution, Bechhofer gives a formula for determining the number of samples to collect that looks similar to our rule of thumb (see also [12], page 275). However, this formula has a different interpretation, and depends on an expansion factor that is a function of k, k, and P in a complicated way. Later works in the area have considered the problem of finding the t best of k populations when $s > t$ populations are selected [10]. A large set of selection and ranking problems involving a variety of distributions and both known and unknown variances is discussed in [3].

In the problem addressed in this paper, the populations have a Bernoulli distribution and the same number of samples is drawn from each population. Sobel and Huyett [24] calculate the number of samples to collect when selecting the best one of k Bernoulli populations with confidence P, and the best population has a chance of success d larger than the second best population. The Sobel and Huyett procedure was shown to be the optimal single-stage procedure [14]. The number of required samples can be reduced by adaptive sampling. A survey of this work is found in [15, 5, 4, 20]. In [4], the theory is extended to selecting the best

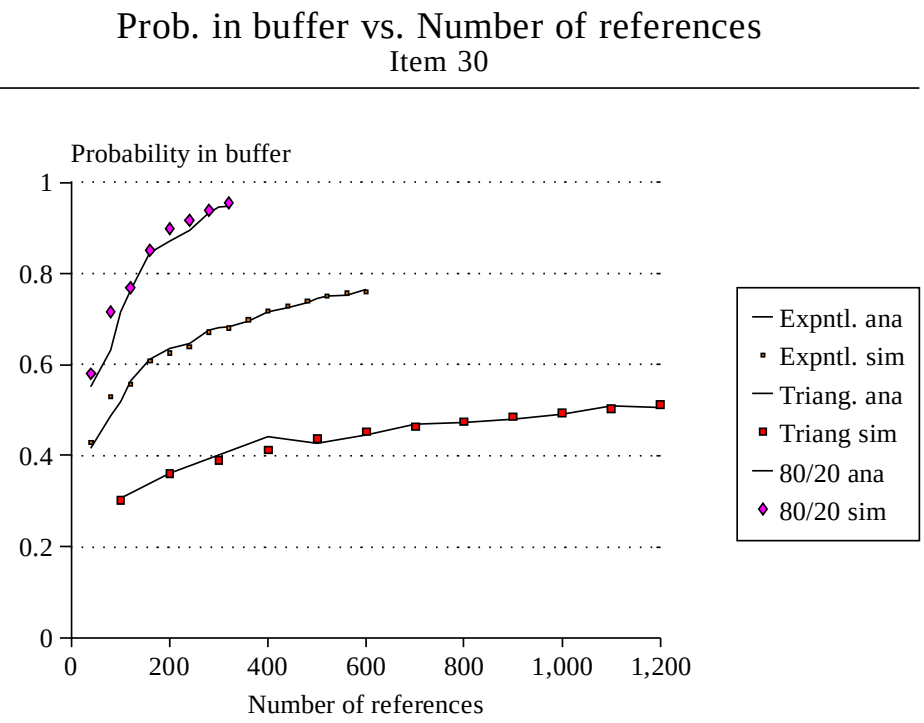

Figure 5: Probability that the 30th hottest item is in the buffer.

t out of k Bernoulli populations (this work shows that choosing the populations with the largest number of success is the best procedure). Jennison and Kulkarni [18] improve the work of [4] to minimize the number of samples that must be collected.

The current work is most closely related to that of Sobel and Huyett [24], since every population (data item) has the same number of samples taken (M). Sobel and Huyett calculated the number of samples to collect by approximating the binomial distributions (the sum of Bernoulli samples) as a Normal distribution and applying the methods described by Bechhofer in [2]. An extention to selecting the t best Bernoulli populations out of K is analogous to the work in [24]. However, we are more concerned with *buffer value* than with finding the t best populations. Though a stopping rule with a similar form to our rule of thumb can be found in [2], it depends on k and t (or N and b) in a complicated way, and the expansion constant has been computed only for small k and t. Since we are concerned with buffer value rather than finding all of the best populations, our rule of thumb calls for far fewer samples.

In Figure 6, we plot the probability that an item is labeled "hot" against the item number for the triangular, exponential, and partitioned distributions when the rule of thumb number of references are taken. Depending on the distribution, the probability that a hot item is labeled "hot" can be fairly low, and the chance of buffering all of the how items is vanishingly low. Yet, the buffer value is fairly high.

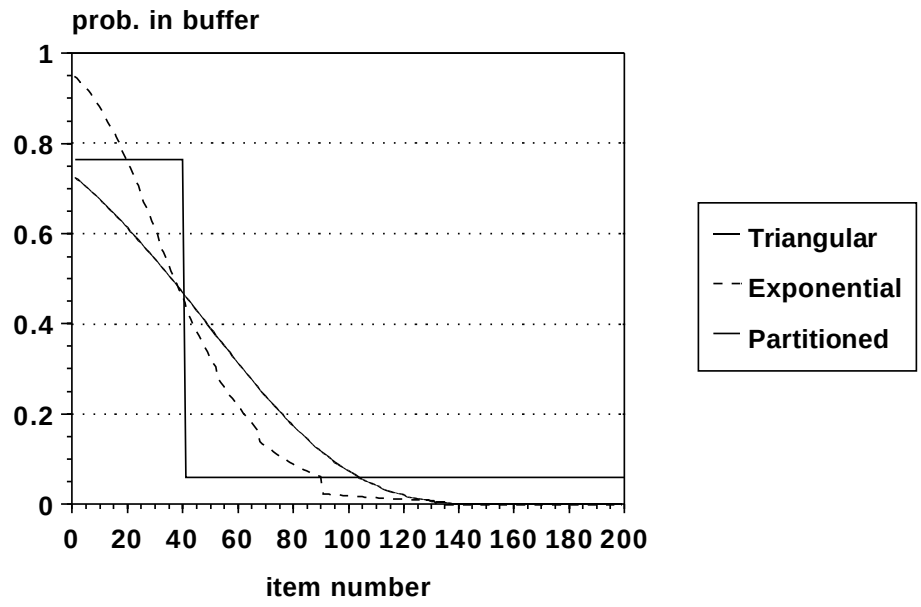

Figure 6: Probability that an item is labeled "hot" when the rule of thumb number of references are collected.

7 Conclusion

Many database optimization activities require an estimate of reference frequencies. However, little work has been done to investigate the quality of the available information about reference frequencies. In this work, we present an analytical model of frequency counting that predicts the quality of a hot spot estimate as a function of the number of references processed. We validate this model by a comparison to simulation results. Finally we present a simple but useful rule of thumb that accurately predicts the number of references that need to be processed in order to make a good hot spot estimate.

References

[1] D.I. Aven, E.G. Coffman, and Y.A. Kogan. *Stochastic Analysis of Computer Storage*. D. Reidel Publishing, 1987.

[2] R.E. Bechhofer. A single-sample multiple decision procedure for ranking means of normal populations with known variances. *Annals of Mathematical Statistics*, 25:16–39, 1954.

[3] R.E. Bechhofer, J. Kiefer, and M. Sobel. *Sequential Identification and Ranking*. University of Chicago Press, 1968.

[4] R.E. Bechhofer and R.V. Kulkarni. *Statistical Decision Theory and Related Topics III, Vol. 1*, pages 61–108. Academic Press, 1982.

[5] H. Buringer, S.M. Johnson, and K-H Schriever. *Nonparametric Sequential Selection Procedures.* Birkhauser, 1980.

[6] E.G. Coffman and P.J. Denning. *Operating System Theory.* Prentice-Hall, 1973.

[7] G. Copeland, W. Alexander, E. Boughter, and T. Keller. Data placement in Bubba. In *ACM SIGMOD Conf.*, 1988.

[8] K.M. Curewitz, P. Krishnan, and J.S. Vitter. Practical prefetching via data compression. In *ACM SIGMOD Conf.*, pages 257–266, 1993.

[9] H.A. David. *Order Statistics.* John Wiley, 1981.

[10] M.M. Desu and M. Sobel. A fixed subset-size approach to the selection problem. *Biometrika*, 55(2):401–410, 1968.

[11] S. Ghandeharizadeh, D.J. DeWitt, and W. Qureshi. A performance analysis of alternative multi-attribute declustering strategies. In *ACM SIGMOD Conf.*, pages 29–38, 1992.

[12] J.D. Gibbons, I. Olkin, and M. Sobel. *Selecting and Ordering Populations: A New Statistical Methodology.* John Wiley and Sons, 1977.

[13] R.L. Graham, D.E. Knuth, and O. Patashnik. *Concrete Mathematics.* Addison Wesley, 1989.

[14] W.J. Hall. The most economical character of bechhofer and sobel decision rules. *Annals of Mathematical Statistics*, 30:964–969, 1959.

[15] D.G. Hoel, M. Sobel, and G.H. Weiss. *Perspectives in Biometry*, pages 29–61. Academic Press, 1975.

[16] M.F. Hornick and S.B. Zdonick. A shared, segmented memory system for an object-oriented database. *ACM Transactions on Office Information Systems*, 5(1), 1987.

[17] S.E. Hudson and R. King. Cactis: A self-adaptive, concurrent implementation of an object-oriented database management system. *ACM Trans. on Database Systems*, 14(3):291–321, 1989.

[18] C. Jennison and R.V. Kulkarni. *Design of Experiments: Ranking and Selection*, pages 113–125. Dekker, 1984.

[19] D. Kotz and C.S. Ellis. Prefetching in file systems for MIMD multiprocessors. *IEEE Trans. on Parallel and Distributed Systems*, 1(2):218–230, 1990.

[20] R.V. Kulkarni and C. Jennison. Optimal properties of the bechhofer-kulkarni bernoulli selection procedure. *Annals of Statistics*, 14(1):298–314, 1986.

[21] M. Palmer and S.B. Zdonik. Fido: A cache that learns to fetch. In *Proc. 17th Int'l Conf. of Very Large Databases*, pages 255–264, 1991.

[22] K. Salem. Adaptive prefetching for disk buffers. Technical Report TR-91-64, CESDIS, NASA Goddard Space Flight Center, 1991.

[23] K. Salem, D. Barbara, and R.J. Lipton. Probabilistic diagnosis of hot spots. In *IEEE Data Engineering Conf.*, pages 30–39, 1992.

[24] M. Sobel and M.J. Huyett. Selecting the one best of several binomial populations. *The Bell Systems Technical Journal*, 36:537–576, 1957.

[25] J.W. Stamos. Static grouping of small objects to enhance performance of a paged memory system. *ACM Transactions on Computer Systems*, 2(2):155–180, 1984.

[26] M.M. Tsangaris and J.F. Naughton. A stochastic approach for clustering in object bases. In *ACM SIGMOD Conf.*, pages 12–21, 1991.

[27] M.M. Tsangaris and J.F. Naughton. On the performance of object clustering techniques. In *ACM SIGMOD Conf.*, pages 144–153, 1992.

[28] C.K. Wong. *Algorithmic Studies in Mass Storage Systems*. Computer Science Press, 1983.