DB2 Buffer Pool Tuning

Statistical Accuracy of Performance Prediction Techniques

A White Paper from Responsive Systems

Dr. Bernard Domanski

February 2002

DB2 Buffer Pool Tuning:
Statistical Accuracy of Performance Prediction Techniques © [1]

Dr. Bernard Domanski[À]
Professor, Department of Computer Science
The City University of New York / CSI

A political scientist wants to know what fraction of the public consider themselves to be Democrats. In this situation, we want to gather information about a large group. Time, cost, and inconvenience, often forbid asking each person whether they are a Democrat or not. In such cases, we gather information about only part of the group to draw conclusions about the whole group.

The entire group of objects about which information is wanted is called the *population*. Individual members of the population are called *units.* A *sample* is a part of the population that is actually examined from which we draw conclusions about the whole population.

A poorly designed sampling procedure can produce misleading conclusions, as the following illustrates. A coil mill produces large coils of thin steel for use in manufacturing. The quality engineer wants to submit samples for detailed lab examination. He asks the technician to cut a sample of 10 squares … Wanting to provide "good" pieces of steel, the technician is careful to avoid the areas of the steel with visible defects … so the lab tests show everything is wonderful … yet the customers complain about the quality of the material received.

Here, the sample was selected in a manner that guaranteed it would not be representative of the entire population. This scheme exhibits *bias*, or systemic error, by favoring certain parts / samples over others. *Random selection* of a sample eliminates bias by giving all units an equal chance of being selected.

A sampling design is the pattern of randomization used to select the sample. The simplest design is equivalent to putting the population in a hat and drawing out a handful (the sample). This is *simple random sampling*, which consists of choosing *n* units from the population in such a way that every set of *n* units has an equal chance to be the sample actually selected.

Consider IBM's DB2 Buffer Pool Analyzer. It selects *n* minutes of activity out of every 30 minutes. ***It defaults to collecting data every 30 minutes, for only 30 seconds, to avoid overhead***. But here, the *n* minutes chosen are at the start of each 30 minute interval. Thus, there is a built-in bias that favors the activity reflected at the start of each 30 minute interval. In point of fact, this sampling design is ignoring all activity that occurs during the remaining *30-n* minutes of the interval.

This is the fundamental problem with fixed sampling designs - there is no way that bias can be eliminated since all of the minutes in the 30 minute interval do not have an equal chance of being selected and analyzed. The collected data will, at best, be representative of only those *n* minutes at the beginning of each interval, and won't be the least bit representative of the remaining minutes in the interval. Thus, in the default collection case of collecting data for 30 seconds every 30 minutes, the sample is representative of

[ß] Dr. Bernard Domanski is a full tenured Professor for the Computer Science department at the City University of New York/College of Staten Island. He can be reached at mailto:domanski@postbox.csi.cuny.edu or by phone:718-982-2843

[1] An excellent statistics text covering probability and statistics is "*Introduction to the Practice of Statistics*", 2nd edition, by David Moore and George McCabe, W. H. Freeman & Co., 1993, ISBN No. 0-7167-2250-X.

30/(60*30) = 30 seconds out of 1800 seconds which is 0.016 or ***1.6% of the activity in the interval, and therefore misses the activity that occurs in 98.4% of the interval.*** At the 5 minute maximum level of data collection for a 30 minute interval, the Buffer Pool Analyzer collects 300/(60*30) = 300 seconds out of 1800 seconds, which is .166 or ***16.6% of the of the activity in the interval, and misses 83.4% of the activity in the interval.*** Collecting data from the beginning of multiple 30 minute intervals, places the sampling bias at the same relative point.

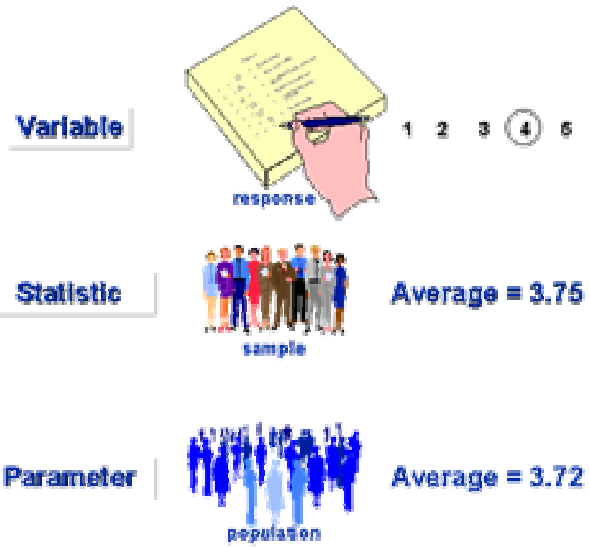
Validity

How valid is it to take samples of activity at the start of each 30 minute interval for just 30 seconds, or even for just 5 minutes? Statistically, validity is related to generalizing. That's the major thing you need to keep in mind. Validity refers to the approximate truth of assumptions, propositions, or conclusions. So, external validity refers to the approximate truth of conclusions that involve generalizations. Put in more pedestrian terms, external validity is the degree to which the conclusions in your study would hold for other measurements collected at other times.

In science, there are two major approaches to how we provide evidence for a generalization. We'll call the first approach the *Sampling Model*. In the sampling model, you start by identifying the population you would like to generalize to - in our case, it's to the 30-minute interval. Then, you draw a fair sample from that population and conduct your research with the sample. Finally, because the sample is representative of the population, you can automatically generalize your results back to the entire population. There are several problems with this approach. First, perhaps you don't know at the time of your study who you might ultimately like to generalize to - but we do! Second, you may not be easily able to draw a fair or representative sample. Third, it's impossible to sample across all times that you might like to generalize to (like sampling during the entire 30 minute period).

IBM's Buffer Pool Analyzer, when collecting samples at the default rate of 30 seconds every 30 minutes, is making the assumption that this 1.6% of the entire interval is representative of the entire 30 minute interval - yet they are missing 98.4% of the interval's activity. Similarly, when collecting only 5 minutes of data at the start of a 30 minute interval, the assumption is that 16.6% of the data of the entire interval is representative of the entire interval - again, they are missing 83.4% of the activity in the interval. While small samples have great appeal in that they minimize the overhead imposed on the system being measures, the generalization that these samples are representative of the entire population are is a poor one. Consider, too, that taking a sample at the same time within every interval greatly increases the possibility of missing the activity of *work* that occurs within each interval, for example, 10 minutes into the interval … or 15 minutes … or 20! Collecting data at the same time within the interval allows one to generalize about what happens *at the beginning of each interval* and not what happens over the entire interval!

Sampling Error and Confidence Intervals[2]



Let's begin by defining some very simple terms that are relevant here. First, let's look at the results of our sampling efforts. When we sample, the units that we sample -- a response time or a service time, or a size, etc. -- supply us with one or more responses. In this sense, a response is a specific measurement value that a sampling unit supplies. In the figure, the person is responding to a survey instrument and gives a response of '4'. When we look across the responses that we get for our entire sample, we use a *statistic*. There are a wide variety of statistics we can use -- average, median, mode, and so on. In this example, we see that the average for the sample is 3.72. But the reason we sample is so that we might get an estimate for the population we sampled from. If we could, we would much prefer to measure the entire population. If you measure the entire population and calculate a value like a mean or average, we don't refer to this as a statistic, we call it a *parameter* of the population.

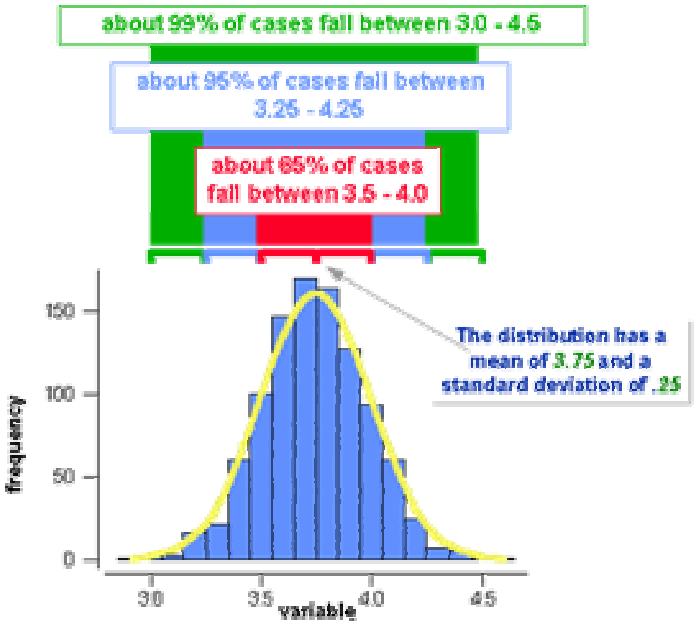
The Sampling Distribution

Sampling Error - In sampling contexts, the standard error is called *sampling error*. Sampling error gives us some idea of the precision of our statistical estimate. A low sampling error means that we had relatively less variability or range in the sampling distribution. But how do we calculate sampling error? We base our calculation on the standard deviation of our sample. The greater the sample standard deviation, the greater the sampling error. The standard error is also related to the sample size. *The greater your sample size, the smaller the standard error. Why? Because the greater the sample size, the closer your sample is to the actual population itself.*

If you take a sample that consists of the entire population, you actually have no sampling error because you don't have a sample, you have the entire population. **This is exactly what *Buffer Pool Tool* from Responsive Systems does!** Samples are taken from the entire interval, and not any sub-interval! In this case, the statistics you calculate are correct for interval - without any error! Again, we must point out the extremely small size of the IBM samples - 30 seconds out of 30 minutes (1.6%) as standard, and 5 minutes our of 30 minutes (16.6%) as the maximum. And again, sampling at the beginning of a 30-minute interval only gives you measurements that are representative of the *beginning of an interval*, and ignores responses that may have been obtained from the middle or end of an interval.

[2] Trochim, William L., "*Research Methods Knowledge Base*", Professor in the Department of Policy Analysis and Management at Cornell University, http://trochim.human.cornell.edu/kb/

The 65, 95, 99 Percent Rule



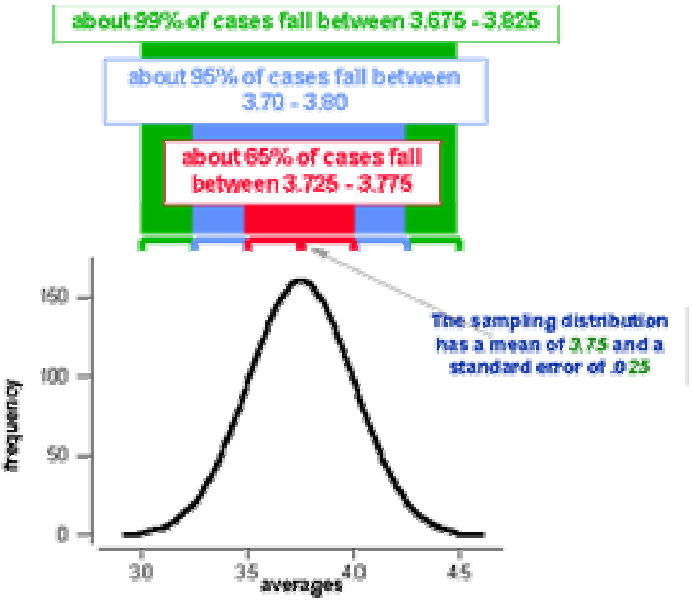
You've probably heard this one before, but it's so important that it's always worth repeating... There is a general rule that applies whenever we have a normal or bell-shaped distribution. Start with the average -- the center of the distribution. If you go up and down (i.e., left and right) one standard unit, you will include approximately 65% of the cases in the distribution (i.e., 65% of the area under the curve). If you go up and down two standard units, you will include approximately 95% of the cases. And if you go plus-or-minus three standard units, you will include 99% of the cases. If we go up and down one standard unit from the mean, we would be going up and down .25 from the mean of 3.75. Within this range -- 3.5 to 4.0 -- we would expect to see approximately 650 of the cases. This section is marked in red on the figure. But what does this all mean you ask? If we are dealing with raw data and we know the mean and standard deviation of a sample, we can predict the intervals within which 65, 95 and 99% of our cases would be expected to fall. We call these intervals the -- guess what -- 65, 95 and 99% confidence intervals.

Returning to IBM's Buffer Pool Analyzer, it is interesting to note that by ignoring the majority of the activity of the interval, ***IBM cannot make a statistical statement regarding the accuracy and confidence of the data they collect over the entire 30 minute interval, or multiple 30 minute intervals***. At best, they can make a statement about the accuracy of the *beginning* of each interval, whether that's only a 30 second beginning, or a 5 minute beginning.

Lets digress for a moment to about **BMC's Pool Advisor**; first, we must state that the Pool Advisor **does not predict performance**. It looks at the 15 minute statistics summary data records generated by DB2, and makes changes to buffer pools based on that analysis. There is no guarantee that those changes will be effective. There is no known *heuristic* that can definitively say that future performance is expected to look like past performance. And there is nothing recoded in the statistics summary data that expresses object access patterns.

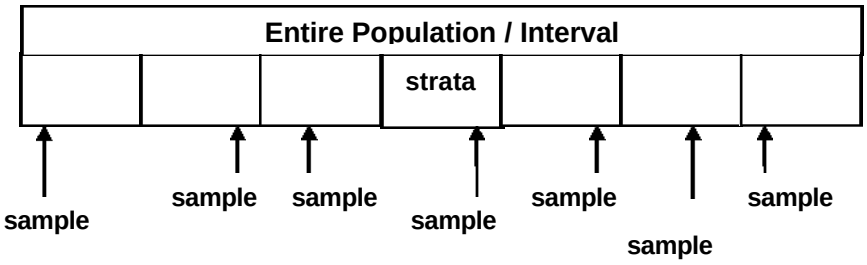
Back to IBM - Now, here's where everything should come together in one great *aha*! Experience it if you've been following along. *If we had a sampling distribution* (not present for the entire interval for IBM's Buffer Pool Analyzer), we would be able to predict the 65, 95 and 99% confidence intervals for where the population parameter should be! And isn't that why we sampled in the first place? So that we could predict where the population is on that variable? We do have the distribution for the sample itself, and from that distribution we can estimate the standard error (the sampling error) because it is based on the standard deviation and we have that. And, of course, we don't actually know the population parameter value -- we're trying to find that out -- but we can use our best estimate for that -- the sample statistic. Now, if we have the mean of the sampling distribution (or set it to the mean from our sample) and we have an estimate of the standard error (we calculate that from our sample) then we have the two key ingredients that we need for our sampling distribution in order to estimate confidence intervals for the population parameter.

Perhaps an example will help. Let's assume we did a study and drew a single sample from the population. Furthermore, let's assume that the average for the sample was 3.75 and the standard deviation was .25. This is the raw data distribution depicted below. Now, what would the sampling distribution be in this case? Well, we don't actually construct it (because we would need to take an infinite number of samples) but we can estimate it. For starters, we assume that the mean of the sampling distribution is the mean of the sample, which is 3.75. Then, we calculate the standard error. To do this, we use the standard deviation for our sample and the sample size (in this case 100 samples were taken, so 100 is the sample size) and we come up with a standard error of .025 (just trust me on this). Now we have everything we need to estimate a confidence interval for the population parameter. We would estimate that the probability is 65% that the true parameter value falls between 3.725 and 3.775 (i.e., 3.75 plus and minus .025); that the 95% confidence interval is 3.700 to 3.800; and that we can say with 99% confidence that the population value is between 3.675 and 3.825. The real value (in this fictitious example) was 3.72 and so we have correctly estimated that value with our sample.



Better Sampling Designs

Sound sampling designs give each member of a population (every possible n-minute subinterval) a known, non-zero chance of being selected. For large populations, the sampling design is sometimes



much more complicated than simple random sampling. For example, it is common to restrict the random selection by dividing the population into groups of similar units called *strata*, and then selecting a separate simple random sample from within in each strata. See diagram above.

Here, the probability of missing activity within a portion of the interval is minimal, thus insuring much more exact information. Stratified random sampling has long been known for this, and has been deployed in many disciplines for many years.

Within the area of performance analysis, stratified random sampling was first deployed in an early MVS monitor (SIMON) developed at Bell Laboratories[3]. Rather than taking samples every *n* milliseconds,

[3] Jacoby, H., Domanski, B. "*SIMON- A Simple Information Monitor for MVS*", Bell Laboratories Technical Report, October, 1977.

SIMON carved up each second into strata, and took a random sample within each strata (see diagram above). Orchard in a special publication of the National Bureau of Standards[4] developed a table that indicated how many samples to take within a second to insure a specific degree of confidence. For example, on an IBM 360 model 168, 7 samples were collected each second (there were 7 strata for each second), yielding an absolute error = .02, and, more importantly, confidence level = 99% (meaning that with 99% confidence, the samples collected reflected the performance over the entire interval. These results were verified with a hardware monitor, which measured all of the activity that occurred each second.

As a result of using stratified random sampling mechanisms in software monitors, the use of hardware monitors was greatly diminished. IBM was, in fact, given a demonstration of the SIMON monitor by Bell Labs personnel in the late 70's, and they incorporated the concepts into MF/1, which later evolved to become RMF.

Summary: Random sampling eliminates bias. Stratified sampling strategies can often insure greater accuracy than fixed sampling approaches. But taking samples at fixed times within an interval has an inherent bias built-in in that activities that occur at the end of an interval (during the 30-n minutes) will not be captured, and thus not reflected in any predictive results generated using the n-minute sample of data collected. A statement regarding the confidence of the data collected cannot be made unless samples are taken across the entire interval, and not just a *too small* sub-interval. IBM's Buffer Pool Analyzer cannot make any statements regarding the accuracy of their predictions; in fact, they do not collect large enough samples to be representative of the entire interval. BMC's Pool Analyzer doesn't even make predictions! Only the ***Buffer Pool Tool from Responsive Systems*** makes predictions that are based on the entire interval under examination.

About the Author …

Dr Bernie's Vitals:
eMail:
domanski@postbox.csi.cuny.edu
Website:
domanski.cs.csi.cuny.edu
(no www. please!)

Comments, questions, and *constructive* criticisms are both welcome and encouraged!

Dr Bernie has been working in IT since the early 70s. He is a well-known figure and has published and presented hundreds of papers worldwide on Capacity and Performance topics. He is a full tenured Professor of Computer Science at the City University of New York/College of Staten Island, and continues to consult to many organizations.

Dr Bernie has been closely involved with CMG for over 20 years; he was awarded the *A. A. Michelson* award in 1992 for lifelong contributions to the systems management field. *Dr Bernie* is a past-President of CMG, and still works tirelessly to help plan the annual CMG conference.

[4] Orchard, R. A., "*A New Methodology for Computer System Data Gathering*", National Bureau of Standards, NBS Special Publication 500-18, Computer Performance Evaluation Users Group, Proceedings of the 13th Meeting, Edited by Dennis M. Conti and Josephine L. Walkowicz; September 1977 NTIS Order No. PB272072.