A New Paradigm in Hit Ratios.

It is becoming more widely appreciated that it is not a good idea to rely on ***hit ratios*** as a method for assessing the efficiency of an Oracle database system. But it is important to remember that an idea is not automatically bad simply because it has gone out of fashion.

Remember that Newton's laws of motion survived for about 300 years before Einstein produced the theory of special relativity, but even today Newton's laws are still totally adequate when used for the right reason and in the correct circumstances.

Although the thoughtless extraction and reporting of ***hit ratios*** is not a sound idea, this paper would like to introduce a hit ratio that could possibly be used to produce a reasonable observation of real-life behaviour.

What's Wrong with Hit Ratios ?

If you take a list of 1,000 numbers and reduce it to just three numbers (and those three numbers are specifically the count, mean and standard deviation) then you have created some useful information. On the other hand, if you start with just two numbers, and reduce them to a single number (specifically their ratio) then you have lost information, and this loss of information is the generic drawback to the hit ratio.

On top of this inherent information loss there is the added, and much more important, problem that there is often a devastating lack of understanding about the underlying measurements being used in the calculation. To clarify this statement, let me quote an abstract of the simplest type of formula typically used to calculate a ***hit ratio***.

```
hit ratio H = 100 * (1 - (measurement P / measurement Q))
```

The trouble is that the measurements P and Q are often just a couple of numbers picked out of some ***V$ dynamic performance view***, and it is often the case that people do not really know the exact significance of what these numbers are measuring.

The classic (though more complex) example of this is, of course, the ***buffer hit ratio***, a.k.a. the ***cache hit ratio***, where one of several measurements used is the number of ***current mode*** gets. However, over the years, the activity associated with a ***current mode*** get has varied dramatically, moreover the actual cost of a ***current mode*** get varies with circumstances, and Oracle performs many actions which are virtually indistinguishable from ***current mode*** gets but does not record them as such in ***v$sysstat***. (You can confirm this last observation for yourself by comparing the activity of the ***cache buffers chains*** latch children with the ***db block gets*** statistic of ***v$sysstat*** when performing different Oracle operations such as update, rollback and delayed block cleanout). So how can a ***buffer hit ratio*** be of any use when there is so much variation possible in the meaning (and accuracy) of the input values ?

The FAN Hit Ratio

Despite my reservations about hit ratios in general, there is one Oracle database hit ratio that I have found to be a reasonable diagnostic tool for real-life applications. This is the ***FAN hit ratio***, which has to be applied at a macroscopic or operational level to an Oracle task. For a given operation (be it query, overnight batch, or anything else) the ***FAN hit ratio*** is calculated (in a fairly typical way) as:

FAN Hit Ratio = 100 * (1 - least (1, calculated response time / actual response time)) (1)

Or FHR = 100 * (1 - least (1, CRT/ ART)) (2)

Note: The FHR is never more than 100, and the presence of the ***least()*** function ensures that the FHR cannot return a negative value.

Of the two input values, the ***actual response time*** (ART) is of course easy to understand and measure. However the ***calculated response time*** (CRT) is more of a problem, and it will sometimes require a fair degree of expertise to be able to produce a reasonably accurate value. In fact I have seen many cases where an apparent performance problem is the result of an over-optimistic value for the ***calculated response time***.

The CRT for an operation represents the optimum time required to perform that operation assuming the database is operationally sound, correctly designed, and the best access path is taken. In the simplest cases the calculation basically requires you to emulate the work done by the ***cost based optimiser*** (CBO) in evaluating the cost of a query, but with the luxury of having perfect knowledge about the system, and plenty of time to do the analysis.

Let me give a few examples:

Example	Calculated response time	Actual response time	FHR
1	3 seconds	12 seconds	75%
2	4 minutes	5 minutes	20%
3	0.12 seconds	0.1 seconds	0%
4	0.001 seconds	0.01 seconds	90%
5	100 minutes	1,000 minutes	90%
6	4 hours	5 hours	20%

Examining the table, you will see that when the actual response time is close to the calculated response time the FHR is close to zero, but if the actual response time is much greater than the calculated response time the FHR starts to climb. For the FHR, zero is best and 100% is worst.

In example 3, we see that the ***actual response time*** is less than the ***calculated response time*** (perhaps Oracle has used a new access method that we did not know about, perhaps some unexpectedly propitious buffering effect has eliminated some expected wait time). So the presence of the ***least()*** function in the formula has given us an FHR of zero.

Examples 4 and 5 demonstrate the pathological problem of ***hit ratios***. In example 4 we have an operation where the FHR is 90% (nominally appalling), however the total elapsed time is only one hundredth of a second and who is going to complain that they have lost a whole nine-thousandths of a second ? However, what if you have to perform this operation 6,000,000 times per day ? In this case we have made a mistake in granularity - we are not really interested in example 4, but in example 5. Does that six million occurrences of the operation take 100 minutes, or 1,000 minutes. A ***hit ratio*** in a vacuum is no good - you need a second number to give you an idea of the scale at which the ***hit ratio*** applies.

Example 6 highlights another limitation of hit ratios - in this case, the FHR is a fairly reasonable 20%. However it applies to a large batch job. If you have only four and a half hours as the window to complete this job, it is ***totally irrelevant*** that your FHR is pretty good - the actual run time won't fit the window.

So what is the FHR ?

Unlike any other of the other ***hit ratios*** invented for Oracle databases the ***FAN hit ratio*** (when used with a little caution, as indicated by examples 4, 5 and 6 above) is a single number that supplies two critical pieces of information. More significantly, though, the FHR is not just a meaningless number, we can equate it with a real-life phenomenon that can be experienced at a gut level by the practicing DBA.

At a human-comprehension level, the ***FAN hit ratio*** is the probability that something nasty is going to hit the fan. Over and above this, however, the ***FAN hit ratio*** is also a pretty accurate indicator of the speed of impact - the higher the ratio gets, the harder the substance hits.

Future Development

Work is still being done to refine the FAN hit ratio. The author has already identified a couple of serious deficiencies in using the ratio as an absolute determinant of trouble. For example, a heuristically based guideline tells us that a ***FAN hit ratio*** of about 50% is likely to be perfectly adequate for most mid-range system whereas for a large-scale, high-throughput, systems a value as low as 3% may not be adequate. Clearly the FHR requires a factor to ensure that its value is automatically increased according to the scale and importance of the system - a Big System (or BS) factor is under investigation.

Furthermore, two variant of the FHR have already been discovered. The first is the so-called ***flying pig hit ratio*** (FPHR) where the ***calculated response time*** (CRT) is replaced by the ***desired response time*** (DRT). Typically, this hit ratio will be seen in management reporting systems where a sub-second response time is required for reports that access tens of thousands of data items. The difficulty here is that the use of the DRT guarantees that the FHR will almost inevitably be very close to 100%

The second variant, known provisionally as the ***friendly user hit ratio*** (FUHR), is derived from the more tolerant use of the ***acceptable response time*** (ART). Recent research suggests that this variant is in fact the best ratio to use as a target for tuning efforts. If you use the original definition of the FHR as your target, it is very easy to be tempted into wasting large amounts of effort trying to achieve the elusive ***perfect zero*** result. (This is of course a consequence of the variant of the Pareto rule that tells us that 80% of the effort will be spent on the last 20% of the work.) However, if you switch from the ***calculated response time*** (CRT) to the ***acceptable response time*** (ART), then the first 20% of the effort may indeed be sufficient to reduce t he resulting modified FUHR to zero. There is, inevitably, a problem here that the author is still struggling to resolve. ACS (Abbreviation Confusion Syndrome) can result in a major indirection of effort; if we substitute the ***acceptable response time*** (ART) for the ***calculated response time*** (CRT) in formula (2) above, we see:

FUHR = 100 * (1 - least (1, ART/ ART))

Careless use of this formula will, inevitably, result in the naïve DBA cancelling the ART with the ART, to produce the result:

FUHR = 100 * (1 - least (1, 1/1))
FUHR = 100 * (1 - 1)
FUHR = 100 * (0)
FUHR = 0

With the modified FUHR apparently evaluating to zero, the naïve DBA could easily be deceived into thinking that their database is running perfectly.

Conclusion

This document is a work in progress, and research is still going on regarding the possible suitability, or safety, of using the ***FAN hit ratio*** and its variants as the basis for a tuning methodology. Any attempt bu the reader to use the ***FAN hit ratio*** on any system, under any circumstances whatsoever, is at the reader's own risk. However, there are several serious points in this document that should not be casually ignored.

Acknowledgements

Mogens Norgaard of Miracle AS (Denmark) for supplying the 15-year old whiskey that made this possible. The audience of the UKOUG Conference 2002 for laughing at the right places.