Business Metrics and Capacity Planning

Ron Kaminski
Safeway Inc.

Yiping Ding
BMC Software Inc.

One of the major goals of capacity planning is to ensure that the business service level objectives are met. Identifying the relationship between Business Metrics of Interest (BMIs) and available system performance metrics is critical to accurate models and forecasts. BMIs represent the real world transactions that drive our business workload's resource consumption. This paper will define the characteristics of a good BMI and give techniques for identifying the best BMI choice from the many possibilities. We will discuss analytical tools, formulas and techniques used, and describe both theoretical ideals and real world cases. We will also show examples on how to use increasingly fuzzy BMIs to make great capacity plans.

1 Introduction - BMIs and Workloads

In the service industry, and in almost every sector of our economy, IT plays a critical role. Business cannot function without acceptable application performance. Understanding the resource consumption causes and trends is key to meeting your service level objectives.

Business activities and system performance are clearly related [1][2]. However, not every business metric that can be gathered has performance implications. In this paper, we introduce Business Metrics of Interest (BMIs) and identify the relationship between BMIs and workload-characterized performance metrics. Workload characterization techniques are a separate topic and are covered elsewhere [3]. Simply put, workloads are groups of processes that consume resources while providing specific functions and services. You can greatly increase the accuracy and the usefulness of your forecasts, models and capacity planning when you forgo total system consumption metrics and compare and contrast different Candidate BMIs (CBMIs) to see which CBMIs drive each workload's resource consumption. Workloads related to business functions should be the focus of your efforts and thus, are the focus of this paper.

In this paper, we will use data from production systems to analyze a problem. Techniques, formulas and examples of how to understand CBMIs, how to deal with the real world issues of incomplete CBMIs, missing data, disruptive secondary processing, sampling and allocation issues, and how to pick the best BMIs from the candidates.

2 Graph and Interpret Your Data!

To examine the relationship between CBMIs and workload performance data, visual or numerical methods can be used. Visual methods offer undeniable power, while mathematical results are precise and can easily be incorporated into an automated process. Ideas from this paper, experience and common sense will help to blend visual and mathematical methods into an accurate picture of current as well as expected future resource consumption.

Our examples will focus on a warehouse support system that currently services three major warehouses; a fourth may be added. The major workload is a database serving both warehouse orders and online (CRM) queries from the marketing folks. The focus will be on finding the CBMIs that drive current workload consumption. We also need to determine what workloads are likely to grow if we add another warehouse, and by how much. Let's look at some CBMIs.

2.1 Graph Workload CPU Utilization and CBMI Over Time

The easiest way to identify CBMIs is to talk to the business. In this example, interviews with business specialists yielded two CBMIs: Order Lines Per Hour and Database Commits Per Hour. Order Lines Per

Hour (Orders) are simply individual order requests for warehouse items coming from retail locations each hour. Database Commits Per Hour (Commits) is a measure of when the system wrote changed data to the database. It is important to note that depending on vendor, database and application design, many order line updates could be serviced by a single commit, and that read-only activities (think about those marketing queries) never cause commits.

Sometimes a simple graph of the workload and each CBMI over time can be helpful. However, our workload CPU resource consumption is measured in units from 0% to 100%, Orders top out above 52,000 and Commits at nearly 26,000. How can we see these on a single graph? The answer is to use scaling factors that convert all values to numbers between zero and one. Since a linear transformation, such as scaling, will not alter the result of a correlation coefficient calculation [2], we can use proper factors/multipliers to bring the values of the metrics into the same range.

Creating scaling factors (***X'***) to compare the workload's resource consumption to each CBMI is easy. For each array (***X***) find the maximum (X_{max}) and minimum (X_{min}) values. Then, for each value of ***X*** in that array, plug in the appropriate values into the following formula [1] to get a properly scaled value (***X'***):

$$X' = (X / (X_{max} - X_{min})) - ((X_{min}) / (X_{max} - X_{min}))$$

Then graph a scaled CBMI and the scaled workload's resource consumption together against time. For clarity of comparison, we put only one set of workload resource consumption and CBMI values on each graph.

Compute a correlation coefficient for each (CBMI, resource consumption) array pair. Correlation coefficients range from a perfect value of 1, meaning every rise or fall in one causes a similar magnitude rise or fall in the other, to –1 where whenever one variable rises, the other variable falls proportionately. Imperfect correlations have values somewhere in between. See [1] and [2] for $\mathbf{R^2}$ correlation formulas.

How did we calculate those correlation values? Since you already have everything in nice columns, add a cell under each CBMI column and use your spreadsheet's correlation function (CORREL in Microsoft Excel). Feed it the workload CPU consumption cells and the appropriate CBMI cells. You can also do it with other tools, or code it yourself.

The values may change slightly based on calculation, rounding and internal number format choices.

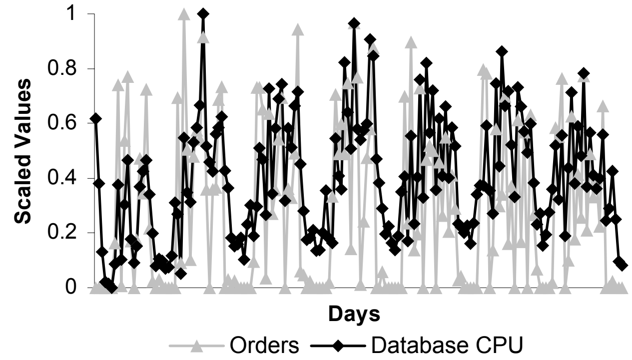

Figure 1, Database CPU and Orders Have a 0.72 Correlation

In Figure 1, you can see that both Orders and Database CPU seem to rise and fall together. While the relationship is not perfect, the correlation factor of .72 is a very positive sign. In the inexact world of sampled performance data and computed business metrics, 0.72 is a very good indication that this performance data (i.e., CPU utilization) and the business data (i.e., Orders) are related.

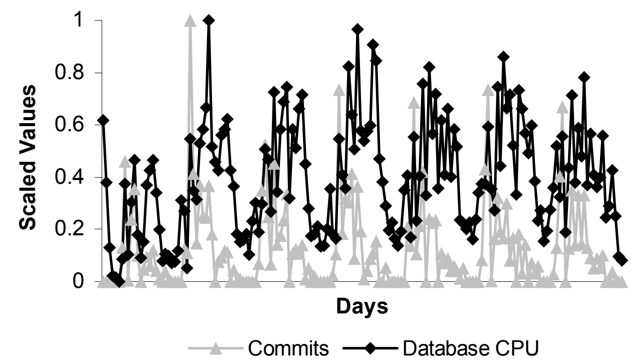

Figure 2, Database CPU and Commits Have a 0.51 Correlation

There is a relationship present in Figure 2, but not as strong as the previous one. Notice how commits are heavy in the mornings (lots of database writes as store managers send in their morning orders) but drop off in the afternoons (the right side of the daily bulges) while database activity typically rises during normal working afternoons (think about those marketing folks issuing read-only queries). So far, "Orders" looks like a better CBMI than Commits.

2.2 Graph Your Workload CPU Utilization Over Your CBMIs

A very powerful way to examine consumption is graph your workload's resource consumption not over time, but versus the CBMI value. In other words, you can have a scatter plot with resource consumption and CBMI as coordinates. If there is a strong correlation between the two, you should see a tight line with a positive slope and one that intersects the origin, meaning zero resource consumption when the CBMI is also zero, like Figure 3.

As discussed in [2], if two metrics, e.g., CPU utilization (***X***) and CBMI (***Y***), have a correlation coefficient of 1, then:

$$Y = aX + b$$

or vice versa, for some constants ***a*** and ***b***, where:

- ***a*** = the slope, i.e. the increase in ***Y*** for each increase in ***X***
- ***b*** = the y-intercept, i.e. how far above the origin our line crosses at ***X*** = 0

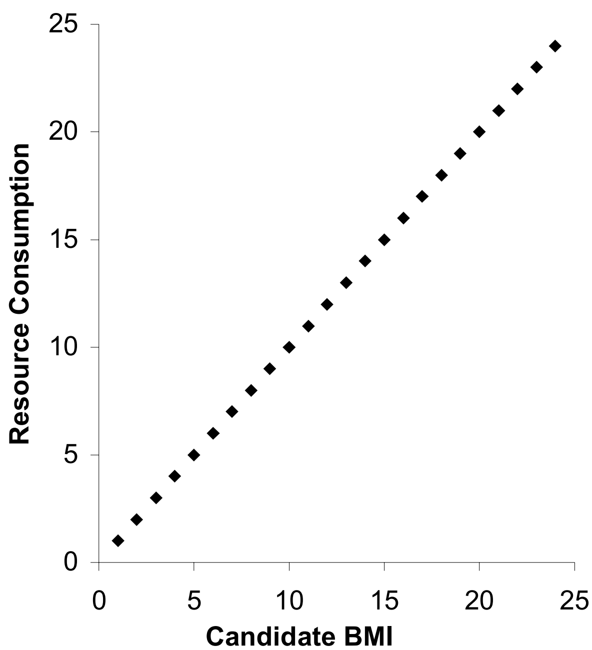

Figure 3, Perfect Correlation

In Fig. 3, you can see that a scatter plot of perfectly correlated metrics is a straight line.

However, there are a few issues with this approach, namely: bulk activity (extreme value) effects, the influence of other undiscovered or ignored BMIs, workload member behavior anomalies, distortions from periods of no CBMI activity that have workload resource consumption, random noise, missing or low value data points and resource or design constraints. How do all these disruptions change our line?

Bulk Activity (Extreme Value) Effects: Bulk activity effects are any activities that can cause a surge in activity that may be normal, but is not typically seen during peak periods when user response time is the paramount concern. Typical bulk activities, like database reorganizations, backups, and table maintenance, can cause huge spikes in consumption, yet typically are not done during peak periods. If these activities do not occur very often, we can throw this data out, as they represent anomalies. If bulk activities happen frequently, especially during peak periods, then they should not be excluded. In fact, this is the cost of doing/maintaining business. We may wish to group them into a different workload to properly represent the bulk effects. Note that this workload may have a very different growth rate than others, and may cause a search for additional CBMIs that describe it.

Undiscovered Or Ignored CBMIs: The usefulness and importance of a particular CBMI is very subjective. Sometimes the users have a favorite indicator, but it may or may not have significant performance or business implications. You need to verify the relevance (correlation.) (Hint: Often DBAs have very accurate transaction logs and the skills to write efficient queries to mine them.) Good thing you made a quick graph and discovered it, eh?

There may be additional CBMI(s) driving consumption (such as the marketing users who load the database with queries) skewing consumption picture. You have a choice; either get another CBMI to represent that population and ferret out the impact or determine that the use is also correlated to the main CBMI (more orders = more data for marketing queries) and just use the single main CBMI as an efficient proxy. This decision will force you to accept a y-axis intercept that rises above zero, but at least you know why. A slight positive y-axis intercept should not be a cause for concern. We are not looking for perfect correlation. We are only trying to identify the workloads that relate to the business activities.

Workload Member Behavior Anomalies And Random Noise: Workload member behavior anomalies are another source of distortion. Whether these are process pathologies like “loops” or “hums” [4], temporary surges or just continuous low level background consumption by some processes in the workload, there is always workload resource consumption, even when there is no CBMI activity. Sampling error may also contribute some additional random noise. This may be normal, and it will tend to raise the y-intercept. From a capacity planning perspective, the random noise will not grow at the same rate as that of the business activities or of the workloads that support the business activities.

Missing Or Low Value Data Points: Sometimes the data is missing, either resource consumption or CBMIs for that period, so we have to throw out all values of performance data and CBMIs in that interval. We also typically throw out "low CBMI" points in fuzzy situations, as the noise is probably swamping the real information, especially near the Y-axis. Make sure that the missing points aren't from critical times!

Resource Or Design Constraints: If you hit a binding limit, or a blocking component other than your measured resource, (such as an I/O subsystem component), your workload can't use more of your measured resource, even if you had lots of work left to do and plenty of other free resources.

Note: This can occur even when you have plenty of machine resources, but application design decisions and program behaviors (e.g. data locking and process threading choices) are limiting throughput. In this case, your scatter plot's slope will flatten out.

Detecting these non-hardware limitations can save you millions, as you avoid buying hardware that you cannot use effectively to improve the performance. Find end user reports of transaction backups at these times, and you can show that the current machine had plenty of resources available. This focuses the efforts of your developers on these issues. In general, a system will have one resource that will saturate first. If this component is saturated, then the overall throughput cannot be improved until the bottleneck is reduced.

What? Me worry? Why should you be so concerned about all these distortion effects? You should find these distortions because they can drastically alter your reasonable capacity plan choices. Each of them can alter the calculated slope ***a***, or the y-intercept ***b***, or both, and thus radically change the results.

All Points Are Not Equally Valid: If you simply regress all data points, you will typically get a line with a high y-intercept value (due to the factors discussed above) and a lower slope. In this case, the slope is even lower because we had a throughput constraint at high CBMI values tugging the high end of the line down.

When business activity increases, different performance metrics will react differently. For instance, random / system activities will not grow at the same rate as the business activity. If you just blindly regress a line based on all points and then extend it to pick hardware, never noticing the factors that distort the line, you risk buying undersized machines, or worse, buy a machine you can't use effectively due to an application design limitation.

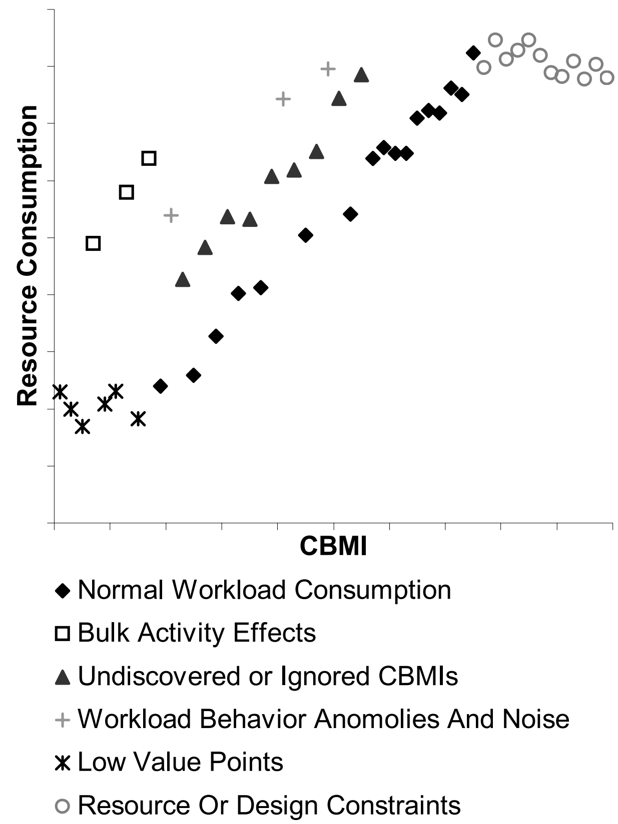

Figure 4, Workload Resource Consumption And A Distortion Bestiary

"Normal" Workload Consumption Is Better, But…: The usual reaction to learning about distortion effects is to use only the undistorted or "normal" points as a planning line. This line's y-intercept may be very close to zero, and the slope is often greater. Assuming that the process of finding "normal" points has forced you to fix any application constraints, you can extend this line and model the correct hardware to support that load, right?

Well, not really. The normal points ignore those application usage surges that always happen, and they also ignore the "business hours" marketing users (hidden BMI) who aren't going anywhere either.

Examine Figure 5, The Proper Planning Line on the next page.

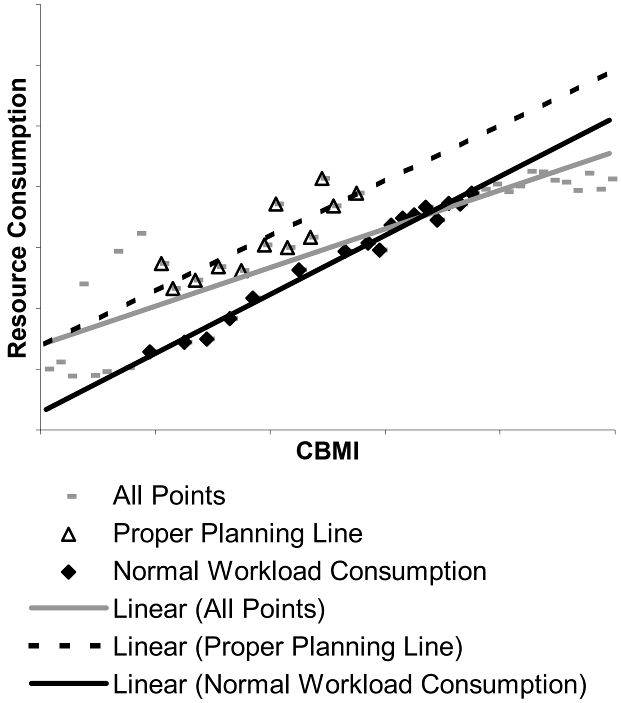

Figure 5, The Proper Planning Line

The Proper Planning Line: It turns out that the best line for planning purposes incorporates the normal “business hours” functions or activities whose response times (i.e. service level objectives) most concern us. If you buy enough machine so that the marketing users are happy despite those periodic application anomalies and warehouse order loads, response times are likely to be great at all other times as well.

Many experienced analysts will find the slope of the proper planning line and then bump it up vertically so that it now intersects their high points, and use that line for safe planning. We will discuss this in more detail in section 3.4.

What About Our Example CBMIs? Orders CBMI versus Database CPU utilization (Figure 6) seems to have a reasonable correlation. We do see some low value CBMI points near the axis, but even with those low value points in place, the $\mathbf{R^2}$ value of 0.5168 is still respectable. This line also passes the visual common sense test: you can see an obvious relationship.

To some, an $\mathbf{R^2}$ value of 0.5168 is not that great, and they are correct in classical mathematical terms. However, in business data processing, you often have to settle for the best of a series of inexpensive, yet imperfect, measures. Perfection can be elusive. The intention of this paper was to highlight methods that help you “make do” with what you often get.

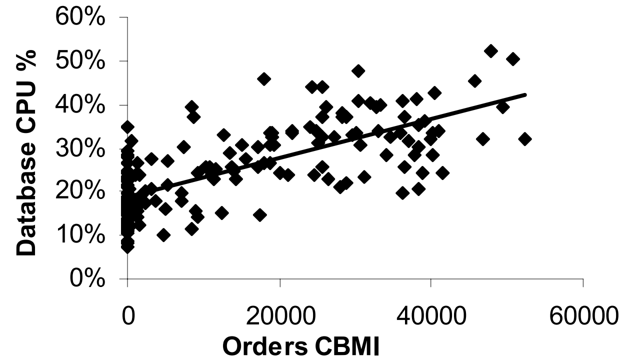

Figure 6, Orders Vs Database CPU, R2 = 0.5168

Commits CBMI versus Database CPU utilization (Figure 7) is nowhere near as pretty. Our $\mathbf{R^2}$ value has dropped to a mediocre 0.2571, which is marginal, considering that 0 means no correlation. You may be suspicious that those few “high CBMI yet low CPU” points to the right appear to be tugging the slope down.

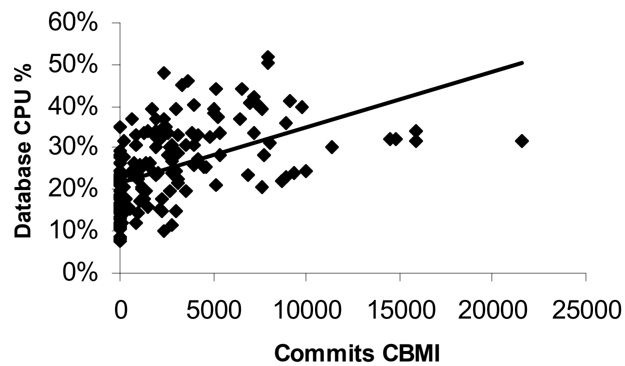

Figure 7, Commits Vs Database CPU, R2 = 0.2571

If our research allows us to eliminate the low and high CBMI points, the resulting fuzz ball doesn’t assist in line fitting. The low $\mathbf{R^2}$ value and inferior visual appearance of “Commits” indicates that “Orders” still looks like the best CBMI.

2.3 What If None Of The CBMIs Are Very Good?

Sometimes you get great CBMIs and your life is easy. Sometimes you try a lot of them and get pretty rotten results across the board. In that case, it may be necessary to keep interviewing business and database specialists, and try a few more candidates. It is really worth the effort, as it is very likely that the BMI you find will come in very handy on any machine related to that function. If you have tested several and they are still in positive, but not wonderful $\mathbf{R^2}$ territory and your deadline is closing in, what else can you do? Look for clustering.

Clustering is a phenomena where a lot of really imprecise (in statistical confidence terms) calculations all cluster around the same value. While each individual R^2 value alone might make you wince, if most of the lines have similar slopes, your data is trying to tell you something. Use the slope, roll it up to cover the peak realistic points and equivocate in your report. Spend the time between this and the next report cycle finding better CBMIs.

Note how similar our two CBMIs slopes and y-intercepts are. This is an example of clustering at work, which should lead to an increase in confidence in your forecast accuracy.

2.4 More Discussion about Keeping and Removing Points

In general, you shouldn't get rid of outlier points that don't fit any of the categories mentioned, especially when those outliers occur frequently within the intervals where business is conducted. Further research may prove that they are real business functions with resource demands that you can plan for.

Once, when one of your authors was reviewing several years of data from a stock exchange, around 17 "high" outliers could not be explained. No one knew why these points were there, but they all represented a surge in trades that might seriously affect machine-sizing questions. After many dead-end interviews, your dedicated author went down to the city library and spent hours pouring through microfilm of the Wall Street Journal for the days surrounding each surge.

Fourteen coincided with federal interest rate changes, two were due to major nuclear capable government changes, and the last was caused when America pursued a significant military endeavor. In each case, these events (interest rate changes, major activities of nuclear-capable governments) prompted surges in trading volume that capacity planners must plan for. If we had just thrown those points away, how accurate would our capacity plan have been? (It would be an interesting study to see if there is a strong correlation between the "color level" changed by the Homeland Security Department and the trading volume.)

If the outliers occur at very large magnitude, but not necessarily in frequency, they will affect not only mean, but also the variance. Some may even exhibit power tail / chaotic behavior [5]. If power tail behavior is detected, then different modeling tools need to be used to capture the high degree of variation.

3 Capacity Planning With Your Slope and Variance Information

At this point we are pretty sure that Orders is our favorite CBMI, so we will promote it to BMI status. In keeping with our ethic, we also check all the remaining workload utilizations against it and determine their slopes and intercepts. These slope and intercepts are useful in examining various growth scenarios.

Our original problem was to determine what would happen if we added a fourth warehouse. Via interviews, we have determined that the new warehouse is likely to generate a maximum of 15,000 additional orders (our BMI) per hour. The vice presidents are asking, "Will it fit?"

The answer that we present should not be based on a simple extrapolation against available resources. What the vice presidents really want to know is if the response times will degrade enough to impact business. Stressing any resource -CPU, IO, and/or network- can lead to elongated response times. We need to examine the response time effects of growing each workload. The best tools for this analysis are queuing theory-based modeling packages, as they calculate not only basic metrics, but also the intricate ways that growing workloads will affect each other's response times.

3.1 Picking A Point To Grow From

Queuing theory models are typically built from sampled resource consumption data categorized as workloads. When you pick the sample period to base your models on, choose a period where there are significant BMIs, and significant representation of all the workloads that will be present during your heavy use periods. In our example, we need a good hour that has both significant orders and marketing queries present. But how do you pick one?

If your BMI is good, and your workloads precise, (i.e. they correlate well with your BMI) in theory any point containing significant use will do. In reality, you should make sure to pick a point where your users are "happy" and working at a normal or sustainable pace. When response times lengthen, humans change their usage patterns, and that will influence the workload resource consumption pattern. These behavior changes may mask true demand, producing a phenomenon called “latent demand”. A model based on a sample taken during a period where latent demand is present runs a serious risk of recommending undersized hardware, because users will revert to higher consumption behaviors when more resources become available.

3.2 Picking Points To Grow To

Our business specialists said that they expect the new warehouse to generate up to 15,000 orders per hour. True usage could be more or less, so a prudent capacity plan will run a series of scenarios from the sample point to some distance beyond their estimate. If the machine has acceptable response time with an additional 15,000 orders, but response times skyrocket at 15,100 additional orders, it would be nice to know in advance.

3.3 Proper Workload Growth Concepts

Now you have good workloads that correlate well with your chosen BMI. You have understood and kept all the valid points that yielded your growth slopes and intercepts, and you have a series of scenarios you want to model. Let’s continue with our warehouse example.

The present warehouse system peaked at 52,294 orders in an hour. If we believe our 15,000-growth estimate, we need to look at how the system will respond to a worst case of 67,294 orders. Common sense indicates that we should probably look at values like 55,000, 60,000, 65,000, 70,000 and perhaps even 75,000. This spread will help us evaluate the response time changes we are likely to encounter in the real system.

But how do we calculate the right amount to grow each workload? There are lots of workload-related theories and practices influencing this choice, particularly the concept of the naïve growth trap [3]. In this paper, however, we will only show you the proper algorithms. Let’s first spend some time on a few graphs that make things a bit easier to grasp.

We went through all that work to get a slope and y-intercept for our workload. Now all we have to do is find our “happy users” point, plug in the numbers and divide, right? Well, look at Figure 8. Our real data points are “fuzz” surrounding the line. Should we really assume that all our growth points are right on the line?

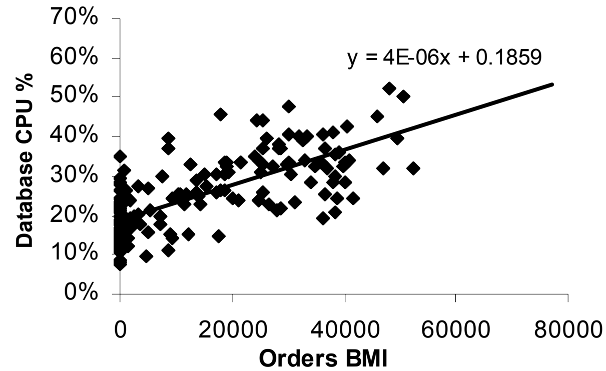

Figure 8, Orders Versus Database CPU Utilization

A favorite visual trick is to simulate some reality in the area of our estimated growth. Add a column to your spreadsheet for estimated growth points. Start with the value of the straight line, and then add some random fuzz above and below. Depending on the random function you use, there are several methods. In Excel, the RAND() function returns a number between 0 and 1, so a formula like:

$$Y = 4E\text{-}06X + 0.1859 + ((RAND() - 0.5) * scale_{factor})$$

where we use a series of ***x*** values in our anticipated growth range. The $\boldsymbol{scale_{factor}}$ is a single cell that we adjust until we get fuzz we like. Then you get a graph like Figure 9.

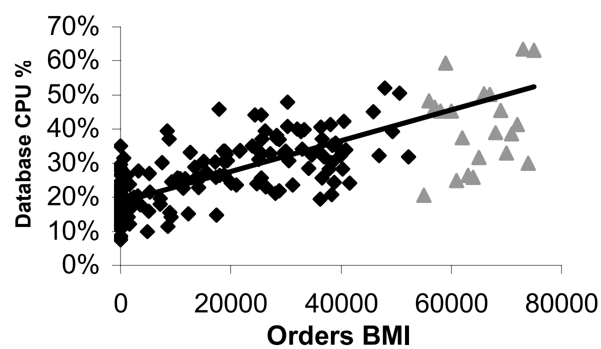

Figure 9, Orders with Estimates

You can just fiddle with it, typing in alternate $\boldsymbol{scale_{factor}}$ values until the fuzz around the line looks nice, but we use statistical probability to guide us. Since 95% of variability occurs within two standard deviations, and 99% occurs within three, setting your $\boldsymbol{scale_{factor}}$ values to produce random points within three standard deviations of the line is a convenient way to be statistically very (99%) confident about your estimate.

In our Excel example:

$$scale_{factor} = 6\sigma_{line\_misses}$$

where:

σ_{line_misses} = the standard deviation of all the differences between observed values and the line value at that BMI.

6? The ***6*** is an artifact of the way we are using the Excel RAND() function in this case. Since RAND() yields a number between 0 and 1, and we subtracted 0.5 to get points above and below the line, we have to multiply by 2 to get our random value between 1 and –1. If you can specify the range in the random function you use, you can skip that. Then we multiply by 3 again to get three standard deviations above and below the line. It is amazing how often that $scale_{factor}$ of $6\sigma_{line\_misses}$ ends up looking great.

The point behind this maneuver is not to generate a "precise random number". (I bet you never thought you'd see that phrase in a CMG paper!) The point is to make you realize that the actual use that you will see in the real production system will vary above and below that line. We generally find a "high BMI", "happy user" sample point near the top edge of the fuzz and then pick similarly placed points in our estimation fuzz and calculate growth.

Why a "high BMI" point? If the users are happy, the less you extrapolate from your sample, the less error you introduce. Given a choice between high BMI and "happy user" pick the "happy user" point, but gravitate to the highest BMI values that you can.

Note: You don't have to do simulations; you can just calculate the numbers as shown in the next section. However, they can be valuable visual aids that may help you explain your results to your users and management.

3.4 Calculating Workload Growth

Let's say that you picked a "happy user" point at 50,618 Orders, where utilization was 50.5%, or 0.505. Of course you would repeat the following calculations for each of your common sense scenarios, but let's focus on a single important workload in the "growing to 70,000 Orders" scenario.

Examine Figure 10.

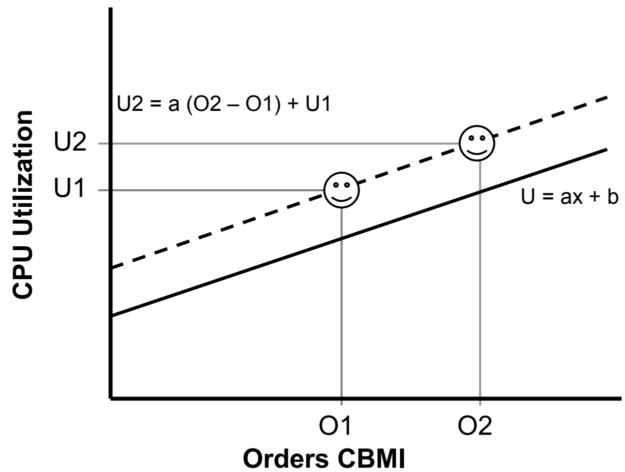

Figure 10, Calculating Workload Growth

What we want to do is travel from our "happy user" point (O1, U1), along a line with our known slope up to our new estimate point (O2, U2), where:

a = the slope of our "all points" line
O1 = our "happy users" sample orders
O2 = our modeled orders
U1 = our "happy users" sample utilization
U2 = our estimated utilization

It is important to realize that the Orders % increase is not necessarily the same as the % increase in utilization. In other words:

(U2-U1)/U1 is <u>not</u> the same as (O2-O1)/O1.

So we need to be careful. We are at risk of making a serious growth calculation mistake. Since what we are trying to find is the increase in utilization we expect for our model, we use the formula:

U2 = a (O2 – O1) + U1

In our warehouse example, numerically, that's:

0.583 = (4E-06)(70,000 – 50,618) + 0.505

Notice that we started with our "happy users" sample point, which is above the "all points" regression line, (actually it is on our previously mentioned Proper Planning Line) and continued upwards along the slope.

This method avoids two common sources of error:

1) the naïve growth trap [3] which occurs when you forget that the y-intercept isn't zero
2) we typically model based on higher points in the variation, the points where all our main

resource consumers are active, not just a point on the line that is likely to be below the peak we will realistically experience.

Armed with our properly estimated database workload CPU value, we need to calculate how much to grow the database workload in our model. In order to properly model growing from 50,618 to 70,000 orders, we need to calculate:

$$(U2\ /U1) - 1 = growth_{percentage}$$

In our warehouse example, numerically, that's:

(0.583/0.505) – 1 = 0.1545 or 15.45%

…which tells us to grow our database workload sample **15.45%** to properly model the 70,000-order scenario.

Note that if we had fallen into the naïve growth trap and divided the orders expected by the orders of our sample (O2 / O1) -1, we would have grown (70,000/50,618)-1 = 0.383 or 38.3%! That is quite a difference, and also the reason why so many capacity plans are way off the mark, even though the tools function perfectly. Remember: To avoid errors, use your BMI to find utilizations and use your utilizations to determine growth percentages.

All you need to do now is repeat these calculations for all your remaining scenarios and all your remaining workloads. Once you have precise workload growth percentages for each workload in each scenario, you can be assured that you will be able to forecast when your "happy user's" mood will change.

4 Summary

Queuing theory-based modeling packages are incredibly powerful and precise tools, but they require precise inputs in order to achieve accuracy. In this paper, we have shown ways to gather Candidate Business Metrics Of Interest (CBMIs) and how to choose the most useful one to drive your planning.

Once you have a favorite BMI, try to frame your capacity issue(s) in terms of the BMI. Test scenarios that include, as well as bracket your growth estimate. Take the time to understand how your workloads respond to changes in your BMI, and you will be rewarded with amazingly accurate growth forecasts and models.

It is easier than it looks!

5 References

[1] Yiping Ding and Kenneth Newman, "Automatic Workload Characterization," CMG00 Proceedings, December 2000.

[2] Yiping Ding, Chris Thornley, and Kenneth Newman, "On Correlating Performance Metrics", CMG01 Proceedings, December 2001.

[3] Ron Kaminski, "*Time Stable Workload Characterization Techniques*", CMG2003 Proceedings, December 2003

[4] Ron Kaminski, "*Automating Process And Workload Pathology Detection*", CMG2003 Proceedings, December 2003

[5] Y. Ding and P. Fiorini, "Maximum Response Time Indicator: Detecting Chaotic Behavior in Performance Data," CMG99 Proceedings, Reno, 1999, pp 300 – 311.

6 Legalese

Work safe, and have a good time!

7 Acknowledgement

Your authors would like to thank Denise Kalm for her painstaking editorial efforts, including going through our paper multiple times. Her suggestions have increased the readability of this paper tremendously.