

# Time Stable Workload Characterization Techniques

Ronald R. Kaminski  
Safeway Inc.

*Workload characterization is the process of organizing computer process resource consumption into groups that aid resource utilization analysis. Great workloads allow the trained analyst to quickly solve performance, growth and sizing questions with precision. However, producing proper characterizations can take a lot of time, especially if they must be built from scratch at the start of each question. The key to a speedy response is pre-work, but how can you do this when you have yet to see the business or technical questions? This session will cover techniques used by experienced analysts to produce general-purpose workloads that have been repeatedly shown to be useful in answering common questions. This paper will also cover workload consumption pattern theory and ways to tune your workloads to increase both forecast precision and audience comprehension. We end with an open wish list to software vendors; in hope that they will provide the additional functions we need to take greater advantage of workloads characterizations.*

## 1 Introduction – Politics Versus Science

The vast majority of IS decisions made are the result of internal political pressures. Often decisions stem more from ego and organizational maneuvering for future plum positions than from a solid basis in scientifically analyzed data. The same basic human need for status that causes midlife crisis sports car purchases can drive management to buy the most powerful machines that they can, and lots of them. There are also economic pressures; many firms base executive pay on span of control and budget size, so there are often financial pressures to spend heavily, and hardware and software vendors are happy to oblige. We also see extreme risk aversion in IS management, so a manager who bought the biggest machine available can't be blamed when the developer's software doesn't fit on it.

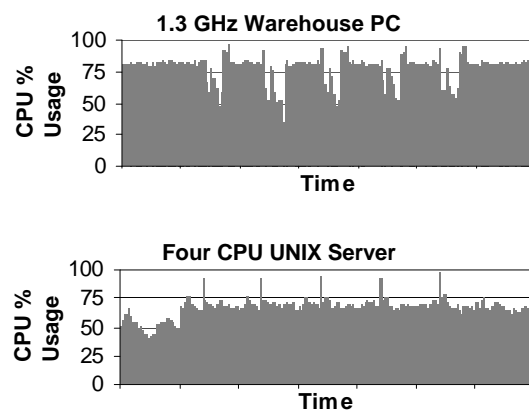
The average non-managerial computer professional has spent a significant portion of their life interacting primarily with machines. Machines are not big sticklers for political correctness, so the computer professional gets little feedback on effective political maneuvers. Thus, they blithely answer only the questions presented by the "schemers," and are often frustrated when this information is used to justify purchases that they feel are ridiculous.

This purchase pattern is unfortunate during boom times when firms can afford mistakes, but can be fatal during lean times. At some point this pattern may lead shareholders to question if management is in breach of their fiduciary responsibilities to the owners. What can we

computer professionals do to break this cycle? How can we help our firms make better decisions?

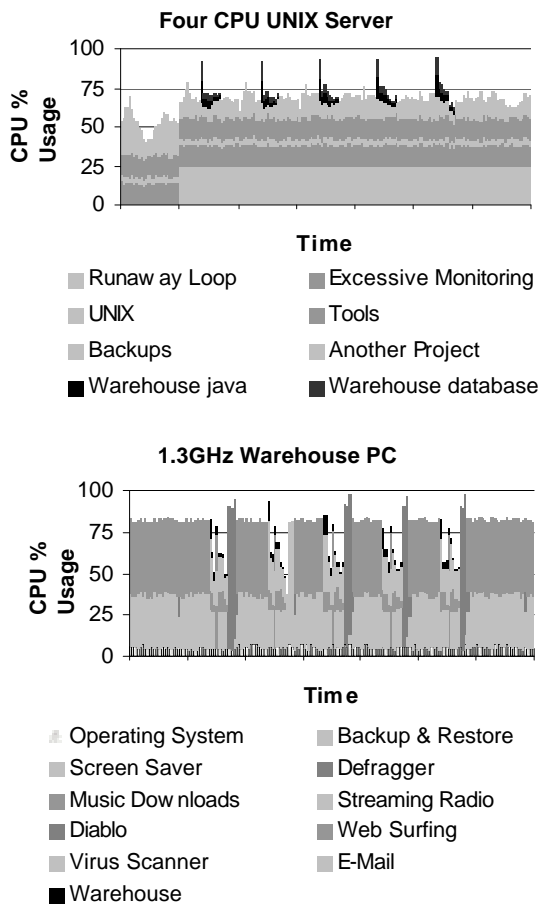
We need to provide better information in more effective ways, and often more information than they ask for. Good business workload characterization is critical to making the right decisions. Workload characterized views of consumption are present in almost every well run IS shop. Let's look at an example of why.

Suppose a development group sends the IS purchasing department a request for a gigantic production UNIX server and hundreds of PC upgrades for the clients to support their warehouse system deployment. They offer these classic CPU graphs (Figure 1) of their benchmark week on their prototype client workstation and UNIX development servers in support of their request:



**Figure 1, Compelling Need For Upgrades**

Imagine how different a response they would get if they offered workload characterized views instead. (Note: Work related to warehouse activities is black and on top.)



**Figure 2, Not So Compelling Now!**

Both Figure 1 and Figure 2 tell the decision makers how much CPU was used on the warehouse development servers, but Figure 2 might lead to better business decisions.

Is this example realistic? An experienced performance analyst will likely confirm that clients often have bloated monitors, projects on machines they didn't know about, developers running things that they shouldn't and an awful lot of surplus processing present on a machine that was supposed to be dedicated to only certain functions. Add in a few process pathologies (loops, etc.) and just finding the real work becomes a hunt.

Without a workload characterized view of consumption, should you really trust benchmark results that only show total consumption?

## 2 Getting Started - What Is A Workload?

A workload is just a grouping of resource consumers. A workload can have zero, one or

many processes and these processes are selected by criteria such as process name, directory location, owner username, or any other differentiator that you can reasonably imagine. Workloads are constructed for a variety of reasons, some better than others, which we will cover in detail later. Generally, a workload is used to summarize the consumption of its members in each period for reporting, or in more advanced tools, for "what-if" modeling.

### 2.1 Workloads That Don't Work!

The availability of a tool to help with the physical process of creating workload-characterized data does not guarantee success. There is often considerable judgment involved, and these judgments are tough for the uninitiated. While the rest of this paper will help you up the workload characterization learning curve, let's take a moment and look at some undesirable detours.

When people first discover workload characterization, they generally split into two camps.

One group dives into workload characterization with missionary zeal. They craft incredibly complex criteria and hundreds of workloads, and the capacity planning and charting machines groan with effort. The other group creates just a few workloads like UNIX, database and users, and produces charts more quickly. Both groups have decisions to make as the workloads evolve, when people add products, functions or new development.

#### 2.1.1 Too Many Workloads!

Our zealot group invests heavily in creating workloads, at least at first. Each node is judged "unique". They fret when any processing slips through as uncharacterized, and become distressed when intricate relationships they defined a month ago are trashed when developers change process names, or directory names change with an upgrade, or any time entropy raises its powerful claw.

Zealots are also prone to "special case mania", in which the same process can end up in different workloads on different machines. This lack of consistency can lead to their audience's confusion and exasperation.

While you can surely see a lot of information on their graphs (if you happen to carry a magnifying glass), the effort to produce and interpret these intricate graphs on an ongoing basis is extreme.

The zealots risk burnout, and rapidly push commercial products to the edge.

### 2.1.2 Too Few Workloads!

The simplistic workload characterizer creates extremely broad workloads, often based solely on username. While it is a step above a simple CPU graph, it is really hard to diagnose whether user Bob is doing real work, has problem processes or anything of much use. The minimalists get little from their efforts, and may soon tire of all the effort they expend for so little gain.

Imagine a database machine with a workload called "database" and another called "other." It isn't much better than a total CPU graph, is it? Perhaps the database should be subdivided into different instances, functions or some other way that is useful to analyze business problems.

## 2.2 Workloads That Do Work

The advanced workload characterizer uses consistent criteria to strike a balance between the two camps. Intricacy is used only when needed, and broad, "sweeper workloads" are defined to minimize the clutter. They also have a stronger weapon - consistency - that makes it easier on them and on the decision makers using their output. Let's explore some of their methods.

### 2.2.1 Beginning Hints

We will describe many good reasons to create specific workloads, but keep these general ideas in mind:

- Use "business based" workload names. If the database is used for warehouse functions, call the workload warehouse, not database.
- Use normal language and short names. Delving into intricate technical descriptions or complex math just makes people uncomfortable. If you have to explain it each time to the viewer, pick a better name
- Avoid defining "nit" workloads. Sure, you've tracked that weird process down and now know exactly what it is. Set a threshold (we use 2.0% - 0.5%), and determine that anything smaller is a nit that doesn't deserve a unique workload.
- Consider your audience and vary your workload characterizations accordingly. It is acceptable to have one set of basic

workloads for graphs that you show everybody and an intricate and complex one that you personally use for analysis or modeling.

- Anticipate the next question and answer it before being asked. Time can be wasted if a killer question is lobbed in and you are not prepared; impatient management might run off and buy something. If you are going to shoot down someone's hypothesis that lack of CPU was the cause of a problem, you'd better find out what really caused the problem before the meeting.
- Cultural differences are real and might affect your workload choices. In some cultures, a username-based workload may be interpreted to cause someone to "lose face". [Foxon 2002] Since our goal is to reduce political problems, not to cause more, consider your choices carefully based on your audience. Any workloads are better than none.
- Be consistent! Always use the same groupings on all similar nodes.
- Use precedence order to decide where to put a process that meets the criteria to be in several different workloads. Rank your workloads from the precise down to the general.

### 2.2.2 The Heavy Hitters

In every firm, there are usually a few well-known monster applications that receive the lion's share of the attention. As a new capacity planner, you will often see these on the top of your "to-do" list. Major databases, payroll applications, integrated accounting automation and customer analysis packages are examples. If the vice president's phone rings when there is a problem, it probably belongs on this list.

The heavy hitters are where you should lean a little in the zealot direction. Often, the effort required to subdivide and allocate the large consumer's consumption will yield the answers to long-standing unsolved problems.

Maybe it would help to see if backups are running in off hours or during the peak? Are there any "well intentioned" but now bloated "home-grown" monitors present, and is their output worth the cost? Any experienced analyst could fill pages with war story examples on this topic. There are often millions of dollars of savings here, so invest the time.

### 2.2.3 But it is just one or a few processes!

Often your heavy hitters have extremely varied or complex functions contained in a very small number of processes and it is difficult to break out consumption reasons; all are indistinguishable from the operating system's point of view. Databases, products like Oracle Financials, SAP and many others make internal metrics available in various forms. The best of these tell you exactly what resources were consumed by function; the least useful tell you function counts and leave you to come up with an allocation scheme. Sadly, vendors that do not provide consumption metrics because collecting them "would impact performance", are often the ones with performance problems.

If you have delved deeply into product manuals, called support and still can't find documentation on the resource cost of a transaction, there are a few tricks to help you figure it out. Perhaps you can find periods where only one type of transaction is present. Divide workload consumption by the transactions completed in that period. I have had success getting the accounting folks to hold all but one type of job for an hour to find these. Often they are really motivated to help, as they are the ones impacted by poor performance during peak periods.

A complex method called factor analysis can sometimes help. Consult a good textbook or do a Google search on "factor analysis" for the gritty details which are beyond the scope of this paper. Basically, factor analysis is the use of mathematical tools that compare resource consumption totals and transaction counts in a series of periods and yield predicted atomic consumption by transaction, with appropriate statistical measures of probability. Software or hardware changes within your analysis period can torpedo your accuracy, so remember not to include these "point source interrupters" in your analysis periods!

Continue with a process of elimination to further subdivide. Remember that your results can shift dramatically when DBAs re-spread tables (usually after you shoot down a machine purchase by pointing out that all the performance problems are due to overloading too few spindles.) Be prepared to re-analyze after upgrades, as time passes, or any time an important decision rides on your output. Often, the usefulness of historical data fades quickly due to the constant stream of small changes in major systems. Focus on recent data. The good news is that you may not need to keep years of consumption data!

In the end, allocation often involves a blend of artistic discretion and science. Lean towards the science.

### 2.2.4 The Usual Suspects

We've all seen movies where the detective faced with a petty crime rounds up the local miscreants for a lineup. After you spend time in your firm's environment, you will find that there are certain processes or packages that show up on many servers, and a problem with configuration or a new version on one machine quickly spreads to many. After finding the same problem on many nodes, you will develop a wary eye when these shuffle into view.

While these may be nits when functioning properly, if they have a habit of misbehaving, they are worth being defined as a workload. We use a Tools workload, where we put things like collectors, monitors, defragmenters, virus scanners, security scanners, and any others that have proven prone to wild consumption spurts. That way, whenever the Tools workload gets big, we know that one of the usual suspects, not business processing, is the reason.

Note, sometimes a usual suspect returns to the path of the straight and narrow. When this happens for a quarter, eliminate the workload.

### 2.2.5 The Joys and Perils of Sweepers

Now you have characterized your heavy hitters, your usual suspects, and most remaining functions. However, there is still that little fuzz of small consumers that aren't in a distinct workload. How do we clean up all that mess? You make sweepers!

A sweeper is often a username-based workload that is far down in the precedence order. On Windows machines the "Administrator" user's processes, and on UNIX machines the "root" process are often used as sweepers. Each operating system has a host of intricate processes that do very basic functions and are always around. Sure, you could enumerate a list and call them "OS Background", "NT" or "UNIX", but every time there is an upgrade, you will have to fix all those workloads. Yuck!

If you are lucky and your firm has username naming conventions that let you divide users into useful piles like "employees", "consultants", "product administration" etc., you can sweep up a lot of flotsam with relatively few workloads. This sounds great so far, doesn't it?

The risk here is that innovation creeps in and suddenly you have a giant sweeper workload. It may be great that the “employee” workload is busy, but what are they doing? Here again, a threshold helps your decision. In general, any time a sweeper rises above 5%, we review it to see if a significant subdivision is possible.

### 2.2.6 “Who Did It?” Versus “What Did it?”

While consulting, you may find shops where there is a rush to blame individuals instead of understanding the technical issue. In general, “What Did It?” is ultimately more interesting, as it leads to calculating the economic business value of a given activity. Keep the username “sweepers” small and you avoid this problem.

### 2.2.7 But I Want to Change a Workload!

If you are new to the workload characterization game, you will be learning a lot. The goal is not to produce a rigidly defined set of workloads the first day. As your business and technical knowledge grows, you will improve your workloads. If the changes are major, remember to spend time with your key audience members prior to large meetings where they might have to interpret your new work. Your first job should always be to make your boss look good in a way that helps the firm.

### 2.2.8 Why Is “Time Stability” Important?

Time Stability is a measure of the resilience of your workloads in the face of changes.

Workloads that remain consistent over long periods reduce the burden on your audience. Corporate decision makers are extremely busy. If they already figured out what the “Tools” workload was last quarter, and they see it again, they can make decisions quickly. If every view that they get from your efforts is inconsistent, you will limit your communication effectiveness.

Also, your time is a scarce resource. If you have to constantly fiddle with your workloads, you limit the number of nodes that you can service, and you reduce the time you can spend on valuable analysis.

Using “Time Stable” workload characterization techniques, you can vastly increase the number of systems that you can effectively analyze. When workloads remain constant, you can more easily develop tools and scripts that leverage your efforts over more machines. This

consistency will also reduce the cycle time between question asked and answer delivered.

## 3 Using Workloads For Analysis

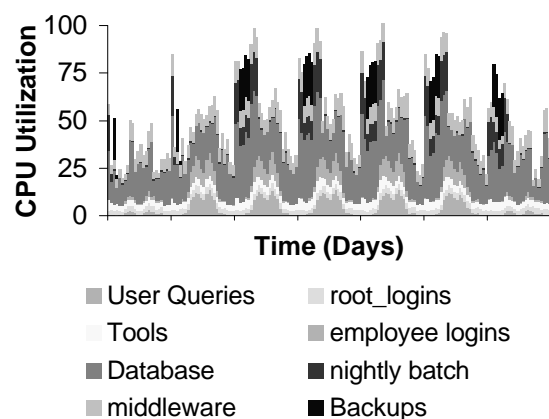
So far we’ve only shown you how to use workloads to create graphs, but there are a lot of other reasons to have them. First, we’ll cover workload consumption theory, or what the graph tells you about how a workload consumes resources. We’ll decide on what is normal. We may even find a problem or two. Then we’ll show examples of how to use the consumption patterns for more accurate estimates and models.

### 3.1 What Business Workloads Look Like

You will graph your firm’s workload consumption, and patterns will quickly emerge. Depending on the lines of business served by the machine you are studying, when you graph based on time you may see the classic two-humped “camel” of a single time zone “office hours” system, or if you serve internet, retail shopping or any other workload that is primarily serving people who are not at work, you will see peaks after normal working hours. Trading firms have their own unique pattern with peaks near market open and close, and industrial sites will have patterns based on shift changes. However, there are many ways to look at workload consumption. Let’s explore a few.

#### 3.1.1 Graphing Based On Time

The most common way people view workload resource consumption is against time.



**Figure 3, A Week On The Warehouse System Versus Time**

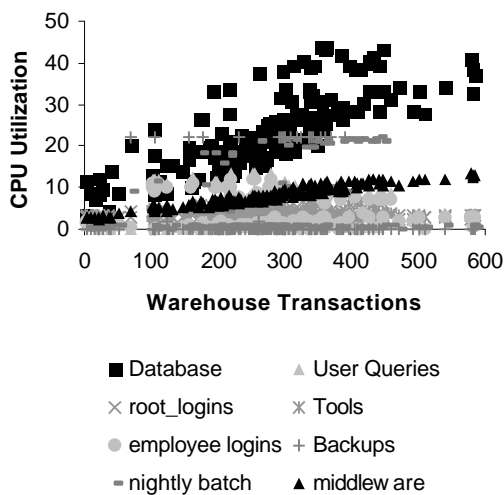
Figure 3 is a week on a system supporting both warehouse users and an office workload with some nightly batch. Note the common pattern of lower consumption on weekends, and how the database workload swells when users add their

queries to the constant drone of warehouse work. There are a few other interesting things going on that we will discuss later.

### 3.1.2 Graphing Based On Business Metrics

A very interesting way to view your workload consumption data is versus a business metric instead of against time. In Figure 4, you can see the warehouse system graphed not by time, but by shipping transactions.

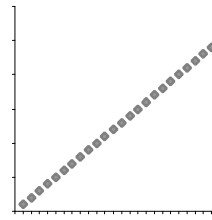
Here we can see several major workloads (database and especially middleware) whose resource consumption is directly related to warehouse transactions. Some workloads, like User Queries, seem to follow a different pattern, probably something to do with when accountant users are at work. We also see some workloads that appear the same or static, no matter what the transaction volume is, like Backups.



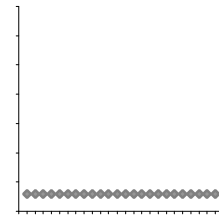
**Figure 4, A Week On The Warehouse System Versus Transactions**

That database workload looks like a line with junk piled on top, doesn't it? You will often encounter these fuzzy lines when you are examining a workload with several independent load sources. You can still see a slope, so we know database load is driven at least in part by warehouse volumes. This type of information will be very useful if your firm has seasonal peaks.

When you graph workload consumption versus business metrics, you will encounter three basic patterns and their combinations.



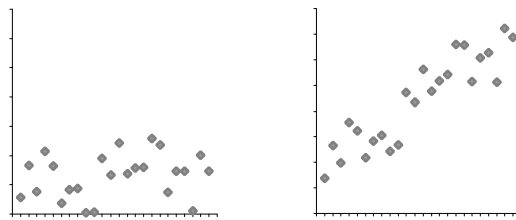
**Figure 5, Linear**



**Figure 6, Static**

Figure 5 is the line you hope for, and you get it when you discover a business metric that corresponds linearly to consumption. You can make great forecasts with these workload-business metric pairs. Figure 6 is also common; it is a workload that just hums along at its own steady pace, no matter what the business volumes are.

Figure 6 looks like our backup workload from Figure 4. A lot of monitoring software and some web infrastructure look like this. Again, you now know that this workload never changes, so you know not to grow it.



**Figure 7, None**

**Figure 8, Combination**

From time to time you get Figure 7. While your eyes will search valiantly, I can assure you that the consumption bears no relationship to the metric you chose for the bottom axis. In our warehouse example, the main database load is driven by warehouse transactions, but human queries are driven by when an accountant hits a key, so they appear random when graphed against warehouse transactions.

Figure 8 is what you will normally see, which is that most workloads have some static component that raises their y-intercept above the origin, and some linear component too. If there is some other load present that is based on a different business metric, you get a fuzzier line. Notice how the combination line in Figure 8 resembles the database workload in Figure 4 that is serving both the accountants and the warehouse folks? A trick for advanced users is to color the dots where the accountants are busy a different color, and you will often see the high outliers pop into view.

### 3.2 Are My Business Metrics Any Good?

Often you will be forced to choose between several different business metrics. You can employ linear regression (if the data should be linear) and check the  $R^2$  value, then, change your workloads, eliminating points that are oddballs or were collected during known periods of resource shortage. There are several intricate mathematical exercises that are described in another paper [Ding, Kaminski, CMG2003] that can help you decide.

Ultimately you will judge whether you have a line or a random fuzz ball. If your otherwise impressive linear line droops at high values, it is often a clue that you are encountering a resource or maybe a design constraint like locking. The good news is that static workloads will always look the same, even if your business metric is lousy!

### 3.3 Extra For Accurate Modelers

If your shop owns or has access to an advanced queuing theory-based modeling package, you can do great “What-if?” analyses of workload growth, upgrade choices and a lot of really precise work. To achieve ultimate precision, make sure you understand how to deal with workloads whose utilization includes a static component.

Suppose we were going to hire 20% more accountants and warehouse workers, and expected resource consumption to rise accordingly. The naïve modeler would raise the appropriate workloads by 20% and see what happens.

The more experienced modeler will remember that all workloads are combinations of static, linear and random components and adjust their growth estimates according to how far these characteristics raise the Y-intercept. Examine Figure 9 for a graphic example.

In Figure 9, we see a workload with a significant static component. Looking at the slope, you may wonder why anyone would pick the naïve growth estimate. But if you never graphed it, and just took the highest use point (79.79) times one plus the expected growth of 20% (1.2) you would get the naïve point of 95.75! Yuck!

If instead you determined the Y-intercept (50) and grew only the linear component of the workload ( $79.79 - 50 = 29.79$ ) by your intended growth (1.2), you

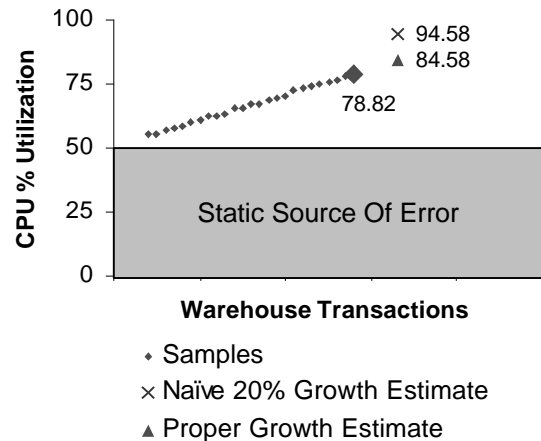


Figure 9, The Naïve Growth Trap

would get the true estimate of 35.75. Remember to add back your static component (50) to get the Proper Growth Estimate point of 85.75. You then divide the Proper Growth Estimate point by the sample to get the proper percentage to grow the workload. ( $85.75 / 79.78 = 1.075$ ) That means you only grow the sampled workload 7.5% for an accurate 20% growth model!

People often make the naïve growth mistake and it leads to inaccurate growth models and over-buying. A great queuing-theory modeling package will yield mathematically perfect yet wrong model results if you feed it bad growth estimates.

I had a great graduate school statistics professor who always insisted that we graph all data and results before and after our fancy formulas, because the eye can detect silliness that formulas hide. You should always graph yours too.

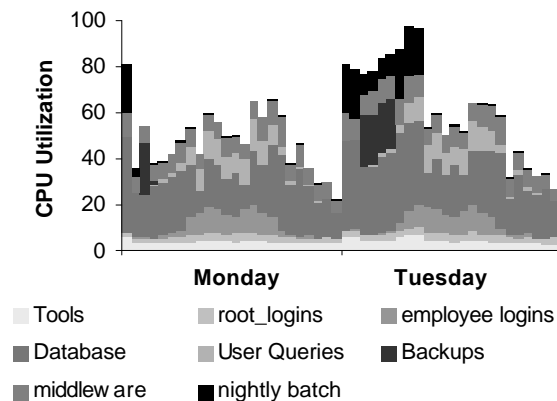
### 3.4 Let's Use Workload Characterization to Solve A Problem!

The warehouse system we've been looking at supports 24 by 7 warehouse shipping activities as well as accountants working regular business hours keeping track of it all. The accountants complain that response time is awful on Tuesday through Friday morning, but is okay the rest of the time. When queried, they tell you that Monday mornings are usually fine, and they suspect that it has something to do with the fact that the warehouse ships far less on the weekends. Management is tired of the complaints and is contemplating a new \$350,000 server offered as the answer to their problems by a salivating hardware vendor.

To start, let's examine a day when the accountants are happy and a day when they are

not, Monday and Tuesday. In Figure 10, you can quickly see that during the first couple of normal office hours on Tuesday, our machine is almost saturated. Lets look at our workloads to see why.

Notice how the backup and nightly batch finish so quickly in the early hours of Monday, because so few database changes happened on Sunday, the “low volume” day before.



**Figure 10, Monday and Tuesday CPU**

Tuesday is a different story. A lot of transactions occurred during office hours on Monday, so the backups had lots of changed data to save, and they ran longer. The nightly batch job also had much more to do, summarizing yesterday’s work for management reporting. Notice how it crawled during backups, as the current machine’s otherwise sufficient IO subsystem clogged with all the data movement. In fact, the probable cause of the accountant’s complaints is that the nightly batch jobs ran two hours into their normal working hours. If you squint really hard at Figure 3, you can see this pattern repeat on Wednesday, Thursday and Friday too!

Note also the gap of low use every evening before midnight. What if we moved backups into that gap? The backups would run faster, as they wouldn’t be slugging it out with the nightly batch jobs for IO bandwidth. Similarly, the nightly batch jobs would speed up, finishing well before the accountants showed up. We don’t need to buy a new machine; we just need to change the start time of the backups!

I’ve encountered this very problem many times over the years, which is why I almost always have a Backup workload. (You can also quickly check if any nodes aren’t being backed up, just check if there is no Backup consumption for extended periods!) This type of analysis can only be done with workload characterized consumption data.

By now, we hope you believe that you can’t live without “workload characterized” views of consumption. How many opportunities for savings will you miss if all you see is total consumption?

#### 4 What to Look For In Vendor Workload Characterization Products

While you can write your own workload characterization programs (shudder), there are a large and ever changing number of vendors selling products aimed at the performance market. While not all metrics lend themselves to these groupings, those that do can be very useful to you. Concentrate your attention on vendor products that offer workload characterized views of collected performance information and provide tools that aid your reporting, analysis and modeling. You might find the following questions useful in your vendor evaluations:

- Ask your vendor how you can subdivide consumption. Favor vendors that can characterized consumption:
  - by username
  - by process name
  - by command line parameters
  - by directory
  - in AND combinations of the above
  - in OR combinations of the above

Better vendors will have most of the list mentioned. I don’t know of any who have them all at press time, but I remain ever hopeful!

- Ask to see examples of workload characterized CPU consumption over large spans of time.
- Make sure that workload characterization is done after the data is collected. Otherwise, you are stuck with your first guess.
- On Unix systems, don’t be content to see total consumption split up only as it is in a “sar” output. I have yet to see where %usr and %sys helped solve a business issue.

Take advantage of the vendor’s strengths, but don’t feel limited to their choices. Great analysis often comes from a synthesis of the strengths of a product and stuff you write yourself. Just remember who has to maintain that mountain of spaghetti code you write!



## 5 What We Wish Vendors Would Provide

Once you've used any vendor product for a while, you will start wishing for things that make your life easier. In many cases, you will tire of waiting and program them yourself. Here is our list of design ideas that make life easier for the busy capacity analyst:

### 5.1 The Über Workload

There are times when a grainy set of workloads really helps show what is going on. There are also times when it is unneeded. It is very difficult to predict in advance just how much detail that your audience may desire. You can of course create a huge number of small workloads, but you may never guess in advance all the ones needed, and your graphs will lean towards zealotry.

We really wish that vendors would create multiple levels of user definable "roll-up" or "über" workloads, which are simply sums of other workloads with user chosen names.

Imagine an "über" workload called Background, which could be composed of sub-workloads like Tools, sweepers like root\_logins, printing, etc. If it stays small, great, if it doesn't, you can drill down for more detail. Imagine a post-consolidation machine with three distinct business functions, you could have "über" workloads for each major function, and then drill in on the monster. With modern web and graphic development tools, supporting "zoom-in" functions like this within the graphics themselves seems like a great idea.

### 5.2 This OR That

While some products have "transaction classes" (which are groups of processes picked by name, directory or command line parameters) that can be combined with each other and username based groupings to produce Boolean "AND" (must have both) relationships, there are times when a Boolean OR relationship (can be in any, don't have to be in all) would be really handy.

Imagine that you have a business function that is the sole function of three clerks. Whatever these three do, you want in that workload. Further imagine that others in the company use "process A" for that same purpose. What you would desire in that case is a workload composed of all use of "process A" by anybody OR any use by the three business function clerks. If you had an AND relationship, you

would only include "process A" when it was run by the three clerks.

You can approach it with groupings of other workloads, but it is messy. Maybe this is just a special case of the "über" workload, but it would still be handy nevertheless.

### 5.3 "Generate On Demand" Web Graphics

Many systems run a daily race to generate thousands of graphs of yesterday's workload consumption between midnight and when the staff appears in the morning. The chance of any graph being viewed is often extremely small, so much of that processing is wasted. We consulted at one very large site that pumped out 28 graphs per node each day for 400 nodes for a year, and only averaged two page hits per month. What a waste!

With the plummeting cost of hardware, and the increasing quality and sophistication of browser-based tools, why not generate only the graphs needed to support the pages people click to, on an as needed basis? If you wanted to get really fancy, pre-make and cache some graphs of nodes on a "hot-list" or that have triggered a process pathology warning.

### 5.4 Workloads That Apply To All Nodes

If you work with these products for any length of time, you will reuse workloads. For consistency's sake, this is a good thing. The trouble is, over time, workloads change. Depending on how you manage your workloads, these small changes can be very difficult to deploy to all your characterization control files. If you are doing it by hand, you are guaranteed to deploy it inconsistently, despite your most earnest efforts.

To solve this problem, vendors need to start to think of characterization as an enterprise-wide activity, or at least an operating system wide effort, not as a single node enterprise. For example, once you determine what a process is on a given operating system, you are likely to want it on all nodes that use that operating system. Vendors need to either 1) add functions that detect when certain workload signatures are present and dynamically apply approved workload characterizations or 2) take advantage of the extremely powerful machines now available and apply all workloads to all nodes.

Without a designed-in method, you will end up with intricate workload characterizations with different precedence orders, different workload constituents or perhaps even missing workloads.

These inconsistencies are unfortunate, confusing and limit your effectiveness.

### 5.5 Workload Color Consistency

Suppose you have worked really hard and maintained consistent workloads across hundreds of nodes. Common programming etiquette is to assign colors randomly to workloads, as long as the colors are unique on a single graph. The same workload on different nodes may be green, red or blue. We have seen repeated instances of confused audiences trying to understand how the red workload on this node is the puce workload on that node. Maintaining consistency by hand is an exercise in frustration, and definitely not scalable.

If workloads were created for an enterprise-wide deployment, you could assign a color at creation, and then puce would always be the same workload. We can hear the users cheering already.

We sincerely believe that sticking to so-called "web-safe" pallets that limit you to a subset of 256 colors is no longer realistic. The 1990s were nice, but the technical limitations of that age should no longer restrict the capabilities of this one. The power of complete information is worth upgrading a browser or workstation that is older than 95 percent of the "dot-coms" still around. We look forward to a vendor deciding to create the competitive advantage that unique colors will bring to reporting and analysis.

### 5.6 Make It Easier

Many people feel that the amount of work required to make workloads is withering. The complexities of precedence order effects (how did that process get in that workload?), the hassles of maintaining consistency, and the repeated instances of analysts in each firm starting from ground zero to find, define and put the exact same processes the exact same workloads is needlessly labor intense.

There are many ways to ease this burden. We hope vendors consider providing libraries of documented shared workloads common to certain operating systems and major commercial

applications. If business metrics were available, wouldn't it be nice if the vendors calculated workload static components and proper growth estimates for us too? Graphic representation of growth choices in the modeling interfaces, and additional workload quality analysis tools would also help.

## 6 Summary

Workload characterizations are incredibly powerful ways to increase the information quality available to decision makers. Whether you use a commercial product, or you program a solution yourself, you owe it to your audience to provide the power of significant, consistent, and business relevant characterized consumption views.

We look forward to large-scale innovation and improvements in workload characterization and reporting technologies in this decade.

## 7 References

[Foxon 2002] Tim Foxon, Metron Athene class, 08/07/2002, Walnut Creek CA

[Ding, Kaminski, CMG2003] Yiping Ding and Ron Kaminski, "*Business Metrics and Capacity Planning*" CMG 2003 proceedings.

A special thank-you to Denise Kalm, the best CMG paper mentor and editor you could hope for. This paper is 200% better due to her efforts.

## 8 Legalese

Any process names, product names, trademarks or commercial products mentioned are the property of their respective owners.

All opinions expressed are those of the author, not Safeway Inc.

Any ideas from this paper implemented by the reader are done at their own risk. The author and/or Safeway Inc. assumes no liability or risk arising from activities suggested in this paper.

Work safe, and have a good time!