

JUMP START YOUR PERFORMANCE CAREER USING ANALYTIC MODELING

DENISE P. KALM

Abstract

What is the career path for a performance analyst? A capacity planner? Most technicians paid their dues in the trenches of systems programming or operations. Having done it all, there can appear to be few interesting challenges left. Move up in your industry to become a strategic planner or a systems architect. Analytic modeling is the number one tool to turn a bit-twiddling performance geek into an advisor to the CIO. This paper will discuss the methodology to make modeling pay off in your career.

Audience

This paper is targeted toward experienced performance and capacity planners looking for a way to regain the enthusiasm and interest they once had in their jobs. Experience with modeling is not a requirement, but those who have used analytic modeling will also benefit from learning how better to communicate the results.

The Career Path Problem

The line between capacity planning and performance management has blurred over the past few years, primarily because of the lack of people to staff the job. Each technician is being asked to do more with less, and to cover more ground. Often, this leads to reactive problem management, which is intensely stressful and generally less rewarding than having the luxury of managing proactively. Pagers and cell phones become leashes instead of tools, and personal time gets invaded. In some cases, boredom is a factor; if you can tune in your sleep, it's time to find something more stimulating. Even for those who truly enjoy their jobs, salary caps and downsizing can be serious issues.

But where do you go? How do you resolve this issue? Can it be resolved? Many opt for a new job, hoping to find more reward in another position, but for performance and capacity planners, this is more problematic. Most came up through the ranks of operations or applications programming. Many moved through a few years in systems programming or subsystem management. The result is that most technicians find that they have "done it all," or at least, tried most of the jobs they felt suited them or were sufficiently interesting to warrant the time and effort. A platform shift might help for a while – zOS to UNIX,

for example, but the methodology is really the same, and the issues don't change that much.

Some opt for a management track, often without a real passion for the responsibilities and challenges inherent in that kind of career move. Management isn't for everyone, and frequently offers less compensation than might have been hoped for. And most aren't aware of how the politics change. Managing upward becomes much more critical than managing the people under you, and many aren't constitutionally suited (or happy) with doing so. Once a move up the ladder has been made, it can be a career-limiting move to attempt a return to the technical track.

In each position, there is an opportunity to become an intrapreneur – to reinvent your own job and tailor it to highlight your proficiencies and expertise, while leaving behind tasks and responsibilities you don't enjoy. Analytic modeling is the key to this for performance and capacity planners, but this means modeling with a difference. The keys to success lie in whom you present it to, and how you present it.

Modeling in Most Shops

What is analytic modeling? It is defined as a mathematical representation of computing systems for the purpose of performance analysis. It represents a snapshot of activity – how long a request sits in the queue and how fast the server delivers service to a request.

Most data processing organizations have one or two people charged with the technical aspects of predicting the future. Many more may gather business projections and create a capacity plan, but

few do the math, and play the what-if prediction games leading to a hardware/software plan.

True analytic (or queuing) modeling can generally only be accomplished with a tool, either home-grown or purchased, as keeping track of the complex math of queuing theory is too time-consuming and prone to error. Many opt for trending, the practice of projecting capacity demands along a line. As processor demand does have a linear relationship with transaction volume, when the transaction mix is held constant, this has some validity, but fails to account for non-linear relationships, such as response time. When linearity cannot be assumed, it is impossible to project into the future where the “knee of the curve” can be found, i.e., where the relationship between transaction volume and response time goes non-linear. In addition, statistical science would hold that in using linear regression (the math behind trending), only values between measured values along the line are truly valid. Projecting beyond those values increases the inaccuracy. So, in most cases, projecting out much beyond the known introduces error.

Trending has been described as, “like trying to drive a car by looking in the rear view mirror....this works on a straight road, but when the road curves, you can be in serious trouble.”

For those who have a modeling tool, modeling is frequently done only once or twice a year for big and mission-critical projects. This has three negative impacts. One – few people actually get comfortable and/or quick with the modeling process when they only get a few chances a year to model. Two – very few people learn how to model, because of the perceived difficulty. Three – most planning/prediction is accomplished via trending, or perhaps some limited stress testing. This increases the risk of inaccurate projections, costly outlays on unneeded hardware and even acquisitions that do not solve the underlying problem. An example of the latter would be buying a larger UNIX server with more power, when all that was needed was 2x the memory, or more disks to spread the I/O activity.

In addition, most modeling exercises are considered only as input to a capacity plan, or perhaps are shared simply with first line technical management. Rarely do key business areas understand the capabilities or have access to modeling results.

Modeling's Greater Potential

Analytic modeling is just too good a tool to only pull it out once or twice a year. It can be used every day to answer the big and small questions that typically are

handled using SWAGs now. If modeling didn't take that long, wouldn't it be valuable to assess the benefit of various tuning options prior to “testing” them in production? Rarely do performance analysts have the luxury of testing recommendations or measuring the benefit of one option over the other prior to installing the change in production; modeling allows you to test it and forecast much more accurately the benefit.

Problem debugging can also be addressed with modeling tools, especially when it is difficult to see what is causing the performance problem. If you suspect memory, just add some more and see how much it helps. Going forward with recommendations, especially those involving capital expenditures, is safer with modeled results.

Although much can be improved through tuning parameters, reconfiguration and re-hosting, a good percentage of response time is generally due to application design considerations. Ideally, performance should be involved in the design phase, and using modeling to assess coding options. But it can also be used after the fact, to test potential improvements, such as reduced I/O. By clearly demonstrating the benefit in response time and/or throughput, the clout to get the applications group to re-code is much greater.

Many performance groups are caught up in reactive performance management. This has some major drawbacks. Reacting to problems is stressful and often, not very rewarding (“Why didn't it work in the first place?”) Furthermore, when problems require hardware or software solutions, favorable pricing is much less available – the vendor is in the driver's seat and he knows it. When you can plan, you can much better control the costs. And basically, reactive management means a lot of monitoring, which can be boring. Most senior technicians like a good challenge and this kind of work ceases to be a challenge relatively quickly.

Even if you are modeling more often, the key is changing focus. Too few of us have the opportunity to work directly with the business partners we serve and this gives us a “glass house” view of things. However, the money comes in via the business units. When you look through this lens in interpreting your work, your value to the company increases in proportion. Pick a key line of business in your shop – one that produces major revenue. Then ask, “What factors in the real world impact the success of this business? What application transactions translate to the key money-making business transactions?” Make business contacts who are willing to discuss their business forecasts with you but then go the extra mile. Bring your understanding of business issues as well

as the technology to build a capacity plan that is more robust, more valid than the “10% a year” growth so frequently offered as a plan. Is the business busier at Christmas? On Mondays? At 6:00 AM? Which time zones? Understanding the business implications make predicting easier and more accurate. But more than that, it translates model results into terms that fit with traditional business cost/benefit equations.

Special Issues in the Distributed World

The same approach applies if your work includes managing NT or UNIX servers. However, the challenges of modeling are a bit greater, which makes having the right tools and a good process even more of a necessity. It is probably even more essential to understand the underlying business, to be able to define “loved one” workloads.

Some of these issues include:

1. No native response time measurement. As the basic business metric is response time, it is essential to find a way to measure this, and particularly, to be able to define changes to relative performance. The OS (operating system) won't help you – the tool selected needs to offer this capability. Without some estimate of end user experience, there is no way to elevate the relevance of your work to senior business management attention.
2. No concept of a transaction. There is no easy way to transform groups of processes into a business function, as there is in the mainframe world. Still, workload characterization is key. The business will not understand concepts like “the ovdr daemon's demand for CPU is increasing.” The better your relationship with the business, the more you will understand and be able to characterize what constitutes a business “unit of work.”
3. UNIX does not equal UNIX. With so many variants, the meaning and even the data elements themselves may not be comparable. Dispatching algorithms and unique processing designs make managing a diverse environment far more complex. It is essential to find a common data gatherer to ensure consistency, accuracy and completeness of the data.
4. OS performance tools can impact the systems they measure. Such tools as vmstat, iostat and sar can add significantly to resource demand when executed at the frequency necessary for good performance data collection. When just the process of observing a system changes the system, the accuracy of the results is impacted.

- In addition, the capture ratio can be as low as 50%, further adding to the data reliability problem.
5. In a multi-tiered environment, network performance is a bigger and more variable component.
 6. The complexity of the interactions adds significantly to the challenge. Many diverse entities are interacting on any given business transaction. Modeling has to take into account this complexity and be able to predict the impact of the proposed change on each entity.

Up-front work characterizing the environment and the workloads is essential, as is creating a process for keeping the information relevant and accurate.

Benchmarking, Simulation and Stress Testing vs. Analytic Modeling

Is modeling the only way to predict the future? No, but it probably is the easiest and least expensive method to deploy. The alternatives are benchmarking and simulation and stress testing.

Stress testing usually involves creating a standalone test environment and employing a tool, such as TPNS to create a workload that can be increased as desired. The issues are as follows:

- The environment is often only a standalone test of a given application. Interactions with other workloads are not included, so this remains an untested issue into production
- There is a substantial effort required to create scripts, manage test files and keep the environment as close to production as possible
- The cost of the environment needs to be considered as well. If shared by development, are the results accurate and repeatable enough to be valid? Can the corporation really afford to create testing environments for all types of hardware in question?
- Often, the database involved is much smaller than production, resulting in a higher likelihood of cached data being retrieved, which would understate the performance impact.
- Prediction capability is somewhat limited. In most cases, hardware options cannot be tested, nor can variable workload growth across all applications.

Benchmarking and simulation are probably the most accurate forecasting methods available. However, the trade-off is that both cost a great deal, both in time and cycles. The coding effort is substantial; opting to test multiple what-if scenarios significantly increases the effort. Often offered by hardware vendors as a

way to evaluate specific site configurations, they don't offer the speed and flexibility of analytic modeling.

Skills and Tools

What tools are necessary? There are a number of options offered by the ISVs (independent software vendors), but the characteristics of the product should include:

1. Read/decode system and application measurement data and convert it into a more useable form. (or in the case of DS (distributed systems) data, collect the data as well)
2. Store the data in a historical database in detail and summary form.
3. Produce reports, graphs and web pages.
4. Predict the performance impact of workload or configuration changes (modeling).

Though only the last involves prediction, it is helpful to find a single product set addressing all your performance and capacity planning needs. For example, after creating a model, it is important to be able to use your regular reporting tools to display model results as well as web publish them. Ideally these tools will have only a minimal impact on system resources (you want to fix the problem, not make it worse) and be easy to use. A top-flight product will require primarily your expertise in performance and capacity planning, not require a whole new skill set. The goal is to offer myriad what-if options in a streamlined, easy-to-use package.

Once the tools are in place, what other skills are required to elevate your predictions (and your visibility) to new heights? The two most critical are knowledge of the business, particularly from a cost/profit basis, and good communication skills. The results are pretty irrelevant if no one knows what you discovered. And as long as data is viewed from only a glasshouse perspective, no one will understand the results outside of the glasshouse. It means approaching your job in a new way – as a part of the service delivery function, instead of technical support, i.e., seeing your role in authorizing credit card sales, assessing and granting loans, facilitating stock transactions.

There are many vehicles to get the message across. A standard, always welcomed approach is the white paper. Key to the presentation is pitching the objectives and results as business problems and solutions and keeping the question and answer right up front. Many senior managers will only peruse the first page – the “high concept” needs to be completely

addressed in one page. (See Fig.1 for a sample paper)

A good white paper will also include the technical and dollar cost analysis, the process used, the assumptions (absolutely critical) and any useful charts and graphs. Costs should include both tangible and intangible items, clearly differentiated – both are important. Keep the format standard to reduce the work effort – one white paper can be cloned into many. The assumptions will often not vary much and having a framework in place shortens the development cycle. Use a white paper when it is impractical or impossible to meet with business partners. As an attachment, it can also be readily passed along, getting the word out there that there is a technician with a “big picture” view of the world who should be included on strategy meetings and project planning focus groups.

Web publishing data is also good, though it frequently doesn't offer an opportunity to add comments and explanations. Ideally, the tool used for modeling has an ability to display reports, graphs and charts to highlight the alternatives proposed. (Fig. 2) In some cases, being able to interactively demonstrate the process and results can be valuable. Rather than the “smoke and mirrors” often offered by tech groups, an interactive demonstration can help to bring the business onto the team, and help them understand the tradeoffs from a technical standpoint.

Almost more important than the communication vehicle is who gets the message. Typically, these analyses live and die in the datacenter. Often, the technician presents results to first and second level DP (data processing) management, but rarely to the business. But who pays the bills? The closer you get to the profit centers, the better your chance of reinventing your career. The people with the money need to understand both the technical and dollar costs of the projects they want to invest in and the changes they plan to make. It's easy to think “economy of scale,” when adding another large portfolio to an existing system, but if there isn't any more capacity, or worse, if the application doesn't scale, the business needs to understand the issue.

Often, the performance/capacity team has no contacts directly into the line of business. There are two approaches to move past this barrier. One – present to the technology asset management committee – the people charged with the actual acquisition of hardware/software – and ask them to invite a business partner. Alternatively, team up with the application group and ask them to include the business. Make sure the information communicated is in a form that allows for it to be passed on, and UP

the management chain. If pitched in terms that make sense to the business, these results can be of interest all the way up to the most senior levels.

Nurture any business contacts you have made, even prior to building a model. This way, if there is any resistance from your own management, you have a valuable ally. If your skills and value are never seen outside the datacenter, you may find you have no future outside the datacenter.

Focusing on the Business

The key here is to ensure that all models and analyses have a business focus. This requires understanding and translating technical concepts into the real world, i.e., which are the key CICS transactions used by customer service agents? What does each transaction do in terms of a business function? Often, overall response time is a meaningless number to the business. There are usually a handful of really critical functions; poorer response time (within limits) is often acceptable for other transactions, and work which constitutes batch processing (on-line print, for example).

Instead of pitching a question in terms of processing resources (how busy is my CPU?) consider what the question means. If acceptable performance during peak periods is achieved, does the CPU busy number really have any meaning? Questions that focus on the business can include:

- Can we make our Web site available to all our customers?
- We would like to merge with another bank of similar size. What happens to our business when we double the volume and number of accounts?
- If the Fed lowers the interest rate, would a "market storm" result in poor response time? Could we find ourselves "eBayed?"
- Do we really need all those UNIX servers?

Bibliography

[Buzen] Jeff Buzen, "Modeling Computer System Performance," CMG'76 Proceedings of the Computer Measurement Group, 1976

[Dixon] Wilfrid J. Dixon and Frank J. Massey, Jr. Introduction to Statistical Analysis, McGraw-Hill Book Company, 1969

[Domanski] Dr. Bernard Domanski. "Simulation versus Analytic Modeling in Large Computing Environments" White Paper from the Responsive Systems Company, 1999

- We need to be able to get into our business application all night long? Can we do it and still get our reports on time?

The other key is to understand that dollars matter. More and more, capacity planners have to consider the dollar impact of their choices. The combination of understanding costs relative to performance has a powerful impact on business partners; you now speak their language.

Where Modeling Can Lead

The kinds of jobs that result from this change in focus can be whatever you choose. Often, these positions will be functions in the business unit, such as Solutions Architect, Technology Integrator, and Relationship Manager. Where you end up often depends on how you style yourself and where you want to be. Don't be afraid to leave the glass house. When working for the line of business, you get close to the decision-makers and become their trusted advisor. And in the business arena, there are few technicians in competition with you; this means you have the opportunity to define the turf you want and expand your technical horizons accordingly. Similarly, in larger corporations, obtaining a position closer to the CIO becomes possible. The goal is to differentiate yourself from other technicians; your increased value should be rewarded with increased pay and status. But mostly, this role can be a lot more fun.

Conclusions

The best job is the one you invent for yourself. Be an intrapreneur, creating your own unique niche in the company. When playing to your own interest and skill set, the rewards are both tangible and intangible. For performance analysts and capacity planners, there is no better tool than analytic modeling to leap past our technicians and become a real corporate player. With the right tools and the right perspective, the sky's the limit for your career.

[Domanski] Dr. Bernard Domanski. "Sizing the Server(s) but Not Knowing Enough: Open System Performance Modeling", CMG'97 Proceedings of the Computer Measurement Group, 1997

[Ho] Eric Ho and Boris Geller, "Performance and Capacity Management of Distributed UNIX Systems: the Glasshouse Approach," CMG'93 Proceedings of the Computer Measurement Group, 1993

[Ho] Eric Ho, "Multi-Platform Performance and Capacity Management" CMG' 94 Proceedings of the Computer Measurement Group, 1994

[Lipovich] Jay Lipovich. "Performance Assurance: A New Paradigm for Performance and Capacity Planning," CMG'93 Proceedings of the Computer Measurement Group, 1993

[Lipovich] Jay Lipovich. "Managing Distributed Systems Performance and Capacity: A Three-Level Approach", CMG'97 Proceedings of the Computer Measurement Group, 1997

Fig. 1 – Sample White Paper

Objective:

Assess the impact of projected transaction volume increases for the real estate loan application on response time for this application and all others sharing the processor. Ensure that service level goals can be met for this and other key applications.

Results/Recommendation:

The increased volume resulted in unacceptable response time increases to several workloads. Several options were explored, but the best price-performer was found to be a migration of the loan application to a processor with more available capacity. This migration would not impact the customer response time of other workloads and would continue to meet service levels established for real estate loan processing.

Technical Analysis:

Several options were explored:

1. No workload moves
2. Move largest application to another processor
3. Upgrade the existing processor

Option 1 did not meet service level objectives. Option 3 involved an upgrade estimated to be \$3.5MM, but this option would have provided acceptable service times. Option 2 met the objective at the lowest cost.

Modeling Process:

(details of data collected, process employed, etc).

Assumptions:

(detailed list of assumptions made, including basic modeling assumptions)

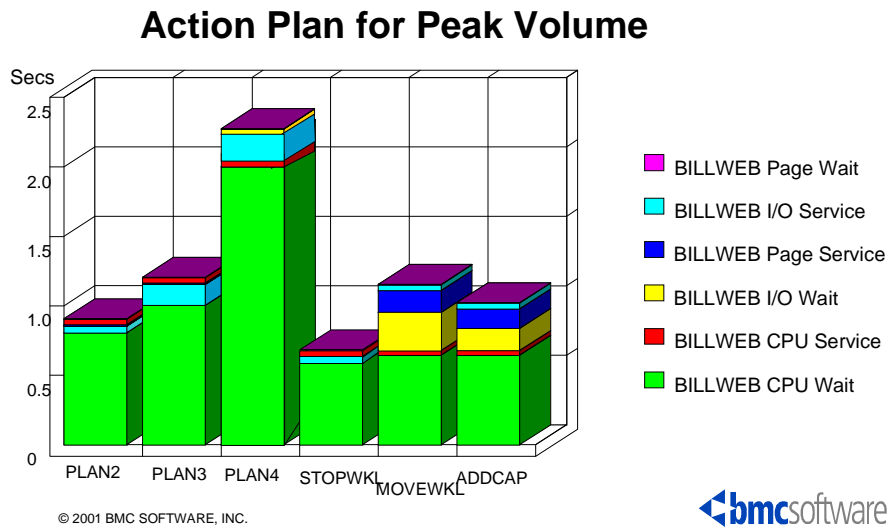
Conclusion:

(a more technical restating of the assumptions)

Charts/Graphs:

(examples of graphs that help clarify the conclusions)

► Predict Impact of Change



27

Fig. 2