Scenario Planning - Part 2

By Adrian Cockcroft - Enterprise Engineering

Sun BluePrints™ OnLine - March 2000

`http://www.sun.com/blueprints`

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303 USA
650 960-1300 fax 650 969-9131

Part No.: 806-4878-10
Revision 01, March 2000

Scenario Planning - Part 2

You can use techniques to help predict latent demand during overload periods, which help define the minimum upgrade needed for saturated components. Using management inputs and modelling alternative scenarios, you can predict the effects of workload mix changes, marketing driven load changes, performance tuning, and hardware upgrades. In part 1, you were shown how to simplify your model down to a single bottleneck. In this part 2, you will be shown how to follow-up a simple planning methodology based on a spreadsheet that is used to break down the problem and experiment with alternative future scenarios.

These articles will form part of the Sun BluePrints™ book titled *Capacity Planning with High Growth Rates* that is scheduled to be published by Prentice-Hall in the fall of 2000. This book will be available through www.sun.com/books, amazon.com, fatbrain,com, and Barnes & Noble bookstores.

Introduction

Start by trending based on recent history, then add forward prediction using business inputs to predict what the future workload should look like.

Trending techniques use a mixture of step functions that track changes in your configuration and a technique sometimes known as Multivariate Adaptive Statistical Filtering (MASF). This process extracts cyclic variations from your data so that you can see the underlying trends. In this technique, analytical models are used to predict the behavior of complex workloads that consist of several classes of user and transaction types.

You should not attempt to build a detailed model of every part of your system. Doing so may be too complex, too hard to calibrate, and may never be finished in time to be useful. Model the primary workloads; make sure you include the ones that make money for your business, like order entry. These need to be maximized.

A Recipe for Successful Scenario Planning

Successful planning must be based on firm foundations. It is quite common for planning to be attempted and abandoned as a technique, because the effort put in was misdirected and the return on investment (i.e. useful planning results obtained) was too low. The recipe published as part one of this article in the February 2000 Edition of <A href=”http://www.sun.com/software/solutions/blueprints/browsedate.html>BluePrints OnLine</A> provides a step by step guide to the process and gives examples of the kind of information that needs to be recorded in each step.

This month, a simple spreadsheet based scenario planning model is explained. It should be very quick and easy to implement, and if nothing else, help you think more clearly about the assumptions you are making over the coming months and years.

Daily and Weekly Variations

The chart in Figure 1. shows most of a week from Tuesday to Sunday on a busy back end server system in terms of transactions per second. Each transaction queries or updates the database, and is a good indicator of the business activity level for this system. You can see that the system never sleeps, but there is a predictable pattern that peaks in the evening during weekdays and falls off a little during the weekend.

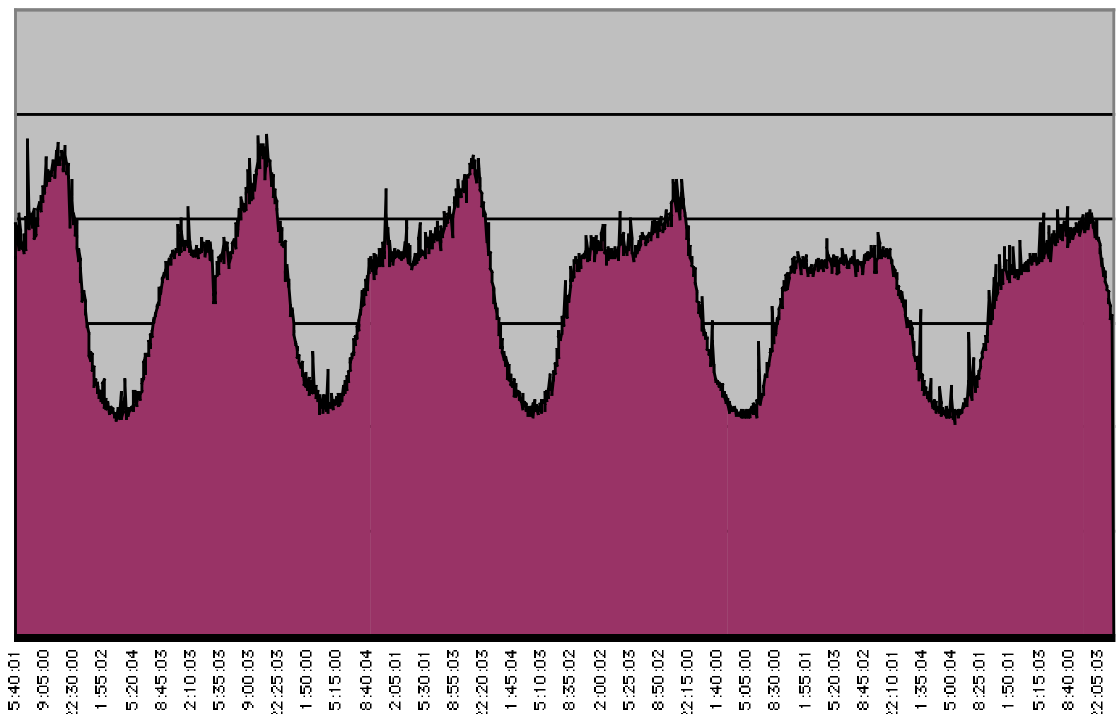


FIGURE 1 Daily Workload Variations Example

The important data is the “prime time” daily peak level. If the prime time is plotted day by day, then a pattern like the factors shown in Table 1 will emerge.

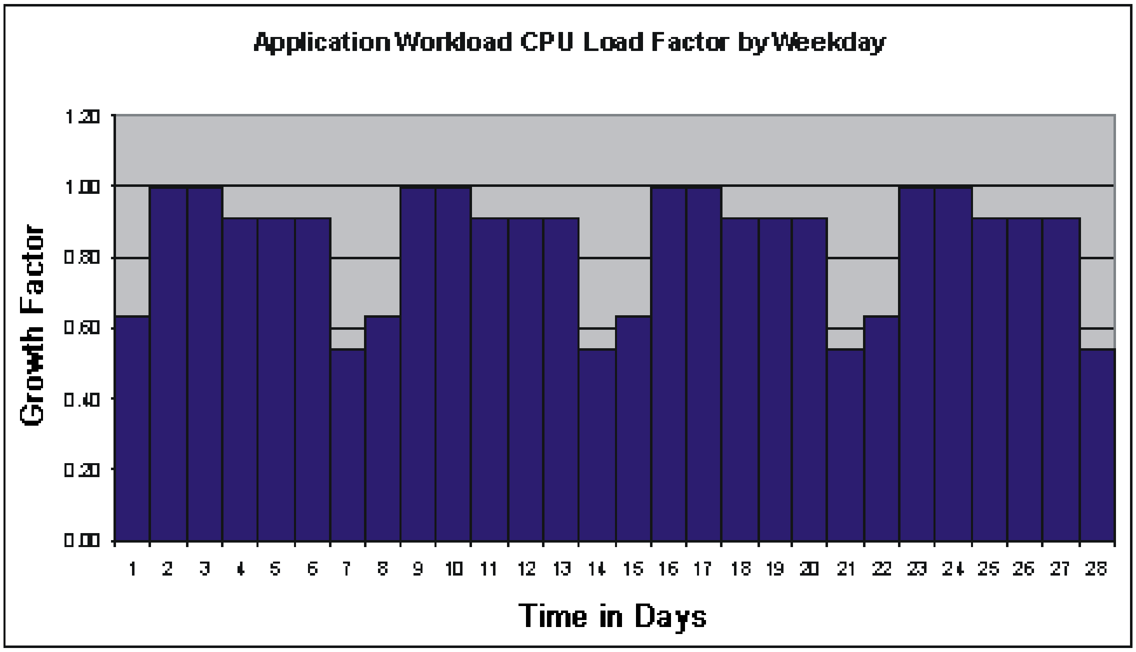
Weekday	Factor	Day	Score
1	0.64	Sunday	0.70
2	1.00	Monday	1.10
3	1.00	Tuesday	1.10
4	0.91	Wednesday	1.00
5	0.91	Thursday	1.00
6	0.91	Friday	1.00
7	0.55	Saturday	0.60
		Normalizer	1.10

TABLE 1 **CPU Peak Load Factor by Weekday**

From the detailed information shown in Figure 1, all you really need is the typical daily peak level. The peak is most important because, good performance during peak loads is required to keep the users happy and make them more likely to visit the site in the future. The actual data from several weeks can be analyzed to extract the typical day to day variation, or enable it to be estimated.

Table 1 is part of the spreadsheet, and you enter your own score levels in the rightmost column to rate each day. These scores are normalized and the factors are generated with the busiest day as 1.0. This data is then charted so you can see the daily variation repeated for four weeks. The chart in Figure 2 shows the peak level day by day with a factor based on 1.0 being a typical day. In this case, Monday is shown as the busiest day, with Friday a little lower, and a quieter weekend. Sundays are often busier than Saturdays, partly because systems located in California are near the end of the time zone; hence, when it is Sunday in California, it is already Monday in the Asia-Pacific region.

FIGURE 2 Daily Workload Variation Factor



Seasonal Effects

As well as variations from day to day, there are seasonal effects from month to month throughout the year. In the rest of this scenario, planning exercise monthly data will be the basis of the plan. Within each month there will be daily variations, but the timescale over which changes and upgrades can be made to a production system, is measured in months. Therefore, the peak load needs to be sustainable in any given month.

If you have at least one year of collected data, then you can use it to generate your estimates for the monthly variations. If not, then you have to come up with estimates by consensus with your colleagues. You can enter these estimates as scores into the spreadsheet in the form shown in the table below. As before the scores are normalized to a maximum of 1.0.

Month	Load Factor	Month	Monthly score
1	0.89	January	8.0
2	0.72	February	6.5
3	0.61	March	5.5
4	0.56	April	5.0
5	0.44	May	4.0
6	0.39	June	3.5
7	0.33	July	3.0
8	0.33	August	3.0
9	0.78	September	7.0
10	0.89	October	8.0
11	1.00	November	9.0
12	1.00	December	9.0
		Scaler	9.00

TABLE 2 Seasonal Load Variations by Month

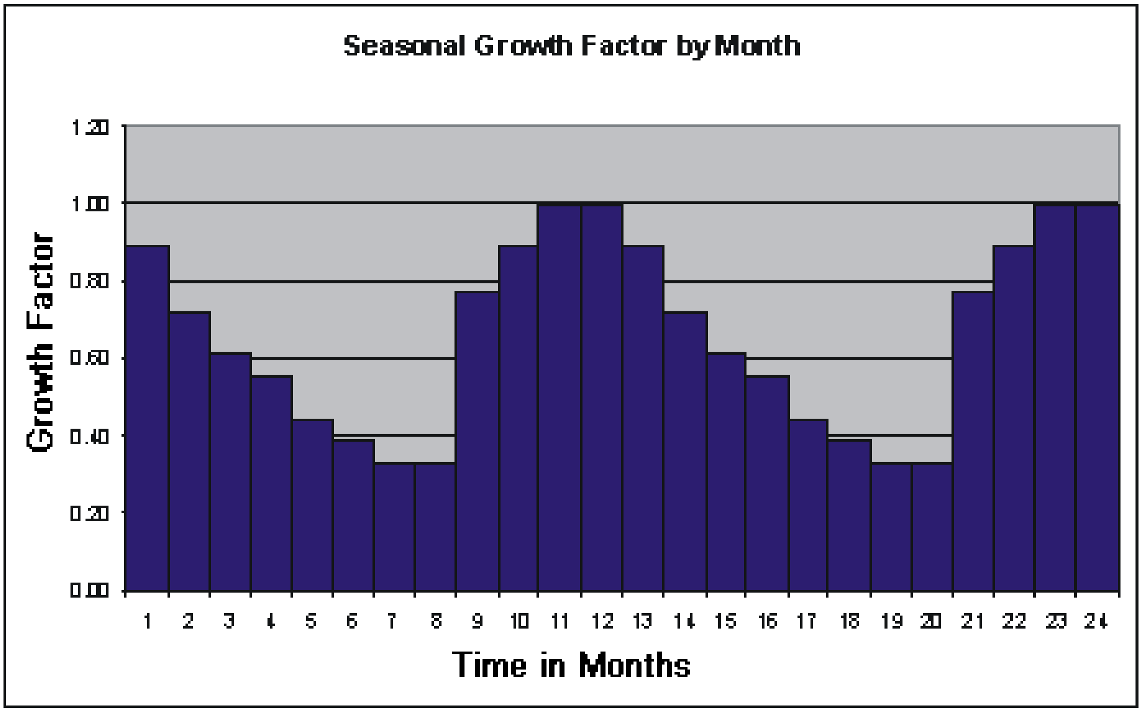


FIGURE 3 Seasonal Load Variations by Month

Two years of example data are shown in Figure 3 starting with January. This pattern is dependent upon external events as well as the nature of the business. It must be determined and calibrated against business metrics, but first other factors must be taken into account. This is shown in the next few charts, which together will form a month by month predicted load level.

Exponential Growth

The growth of the internet combined with growth in awareness of your site causes an exponential growth in load levels. This can be easily expressed in two numbers as a growth factor over a time period. For example, growth might be expressed as 50% per quarter, or doubling in a year. The spreadsheet takes these two values and calculates the exponential growth factor, the doubling period, and the monthly factors for the next 24 months.

TABLE 3 Exponential Growth in User Activity

	Exponential Growth		Parameters	
Factor	**0.135155036**	**Duration**	**3.0**	**months**
1	1.1	Growth	1.5	times
2	1.3			
3	1.5	Doubling period	5.1	months
4	1.7			
5	2.0			
6	2.3			
7	2.6			
8	2.9			
9	3.4			
10	3.9			
11	4.4			
12	5.1			
13	5.8			
14	6.6			
15	7.6			
16	8.7			
17	10.0			
18	11.4			
19	13.0			
20	14.9			
21	17.1			

TABLE 3 Exponential Growth in User Activity

	Exponential Growth	Parameters		
Factor	**0.135155036**	**Duration**	**3.0**	**months**
22	19.6			
23	22.4			
24	25.6			

This growth rate looks like Figure 4 when it is plotted.

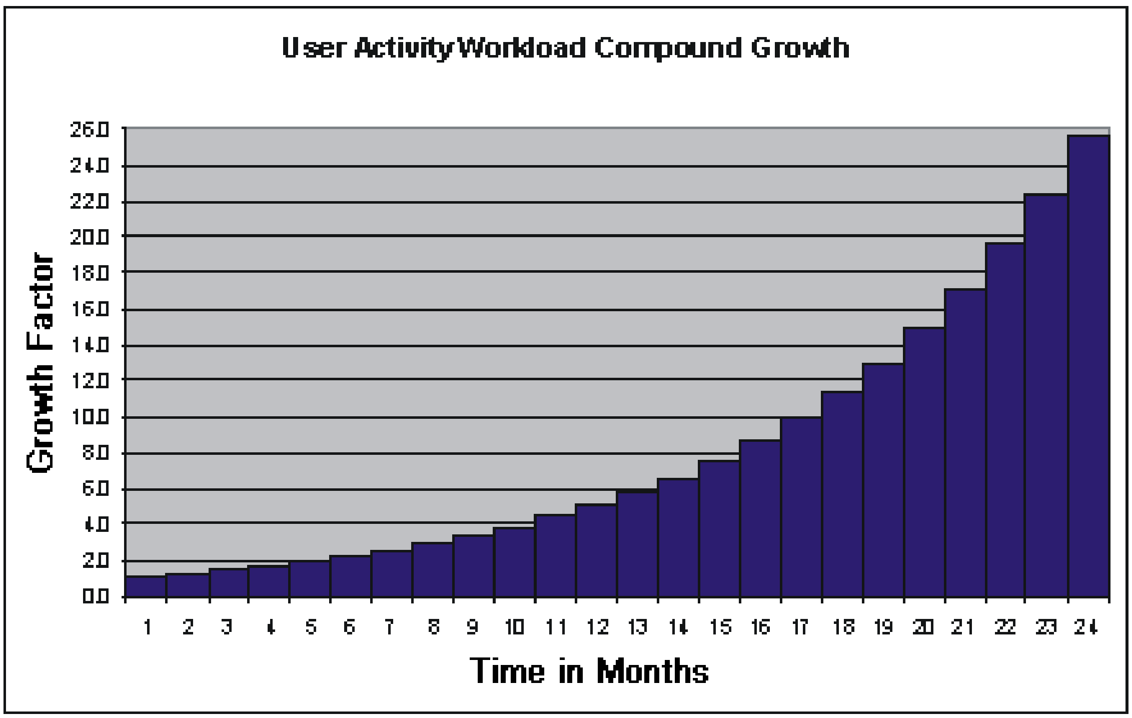


FIGURE 4 Exponential Growth Rate

There is an underlying high growth rate in the size of the total Internet marketplace, and in the activity of Internet based users. Several large Internet sites are seeing business grow at a rate of about 50% in every three months, or a doubling period of just under six months. This translates into an exponential growth as shown above. After two years, the monthly load level is about twenty five times higher than at the start. Your growth rate will vary and it may not continue at this rate for this long, but some kind of growth model needs to be estimated and then calibrated against experience over time.

Marketing Events

There have been many examples of companies that run a marketing campaign that is too successful in getting new users, and overloads their systems. To add this effect into the future growth plan, a spreadsheet based model has been developed that parameterizes the expected increase in load. The parameters are expressed in a form that should be similar to the justification for the campaign generated by the marketing people. These parameters include: the duration of the campaign, a scale factor related to the impact or reach, a residual level which is the long term increase in use load due to this campaign, and a delay that sets the starting point for the campaign in the model.

Month	Factor	Parameters	values	units
1	1.00	Duration	3.4	months
2	1.00	Scale	3.0	times
3	1.00	Delay	4.0	months
4	1.30	Residual level	1.2	times
5	1.54			
6	1.78			
7	1.86			
8	1.76			
9	1.58			
10	1.41			
11	1.30			
12	1.24			
13	1.22			
14	1.21			
15	1.20			

TABLE 4 Marketing Campaign Load Boost Factors

The boost can be seen more clearly in Figure 5.

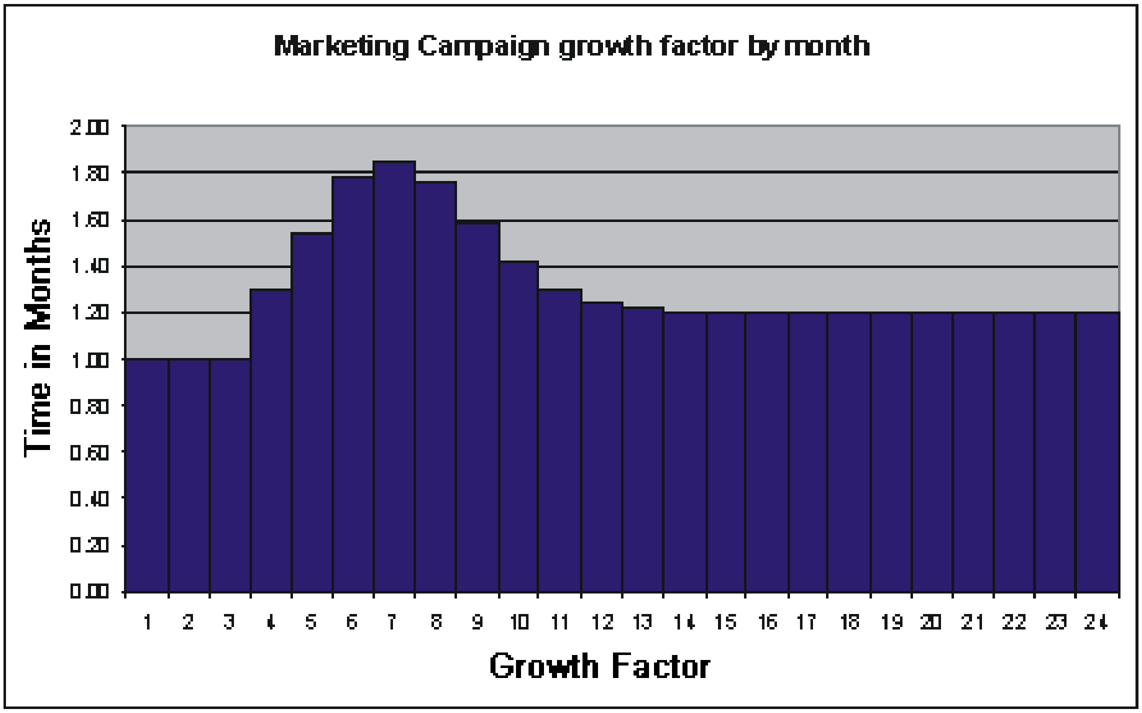


FIGURE 5 Marketing Campaign Boost

A marketing campaign is shown starting in the fourth month, to counteract the expected seasonal drop in load level. It continues for four months then the memories fade away and a residual increased level of activity is seen. The short term and residual gain should be something that the marketing department uses to justify the campaign in the first place, so its impact can be modelled in the same way as its impact on the business. The campaign is not repeated in the second year, you will see in later charts why it was not needed. In practice, many smaller marketing boosts may be modelled rather than one big one.

Efficiency Variations

The model is based on a constant average CPU usage per transaction. Large changes in the application or transaction mix need to be accounted for in the scenario plan. Ideally, performance tuning of the application will reduce the CPU used per transaction every few months so that a schedule for expected (or desired) changes in the application efficiency can be laid out as part of the model and recorded in the spreadsheet.

Month	Factor
1	1.00
2	1.00
3	0.80
4	0.80
5	0.90
6	0.90
7	0.90
8	0.70
9	0.70
10	0.55
11	0.55
12	0.55
13	0.55
14	0.50
15	0.50
16	0.50
17	0.30
18	0.30
19	0.30
20	0.20
21	0.20
22	0.20
23	0.15
24	0.15

TABLE 5 Application CPU Usage Per Transaction

These factors are plotted in Figure 6.

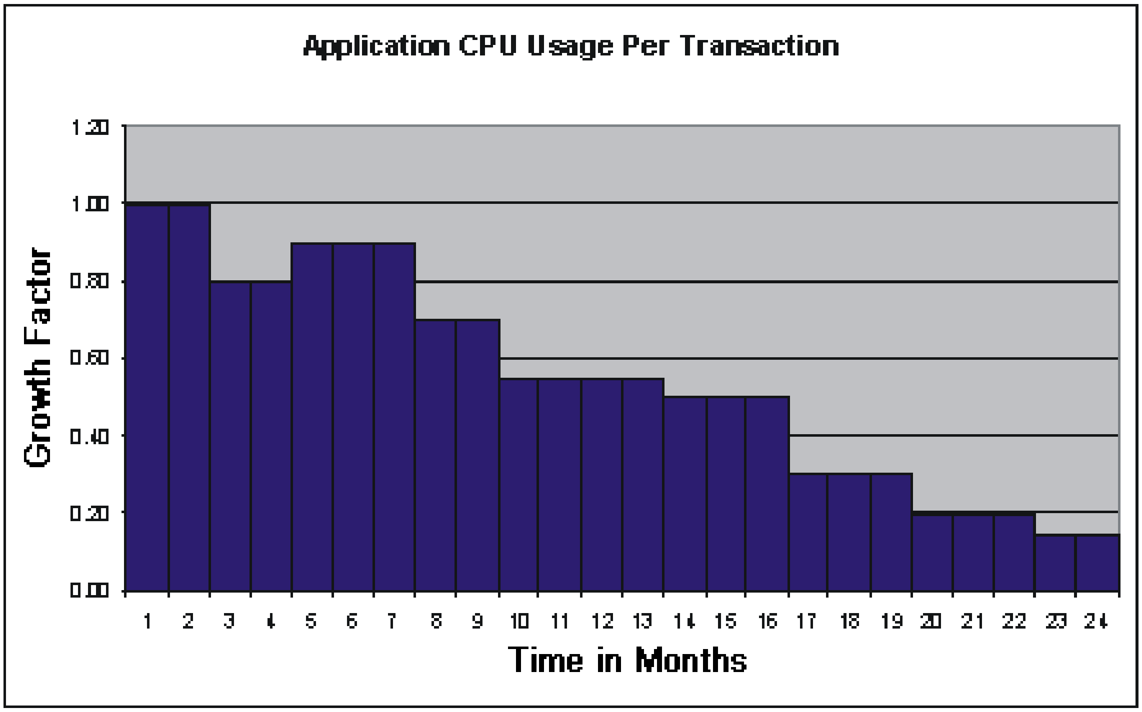


FIGURE 6 Chart of Application CPU Usage Per Transaction

To counteract the increased load levels the application is tuned and the hardware is upgraded. The effect of tuning can be modelled as an efficiency step function, where the growth factor is shown reducing as the load level decreases on a better tuned system. In the chart shown above, database tuning or application software upgrades give the first two gains, then there is a reversal as a new more complex user interface is introduced. This makes the users happier but adds to the load. After a while, several more tuning improvements are made until after two years, then the computer resources used per transaction are reduced to 15% of that at the start.

Capacity Increases

Hardware vendors can supply capacity increases in the form of more efficient operating systems, more CPUs, faster CPUs, or new generations of systems. This causes a series of steps that are captured in the spreadsheet and shown graphically in Figure 7. The performance factors are estimated based on experience with the application in use and standard industry benchmarks.

Month	Factor	Comment	Performance	Utilization
1	1.0	CPU E6K 12x333MHz+4MB	1000	71%
2	1.0		1000	92%
3	1.0		1000	51%
4	1.0		1000	69%
5	1.2	CPU E6K 16x333MHz+4MB	1200	71%
6	1.2		1200	82%
7	1.2		1200	84%
8	1.4	Oracle Upgrade	1400	60%
9	1.4		1400	145%
10	1.6	CPU E6K 20x333MHz+4MB	1600	117%
11	1.6		1600	139%
12	1.8	CPU E6K 24x333MHz+4MB	1800	135%
13	1.8		1800	134%
14	2.0	Solaris™ 2.6 upgrade	2000	101%
15	2.0		2000	98%
16	2.0		2000	101%
17	2.0		2000	56%
18	2.0		2000	56%
19	2.0		2000	55%
20	3.0	CPU E10K 40x400MHz+8MB	3000	28%
21	3.0		3000	74%
22	3.5	Solaris 7 upgrade	3500	83%
23	3.5		3500	81%
24	3.5		3500	92%

TABLE 6 Hardware Upgrade Capacity Increase Factors

For this example, some system configurations and performance factors were invented.

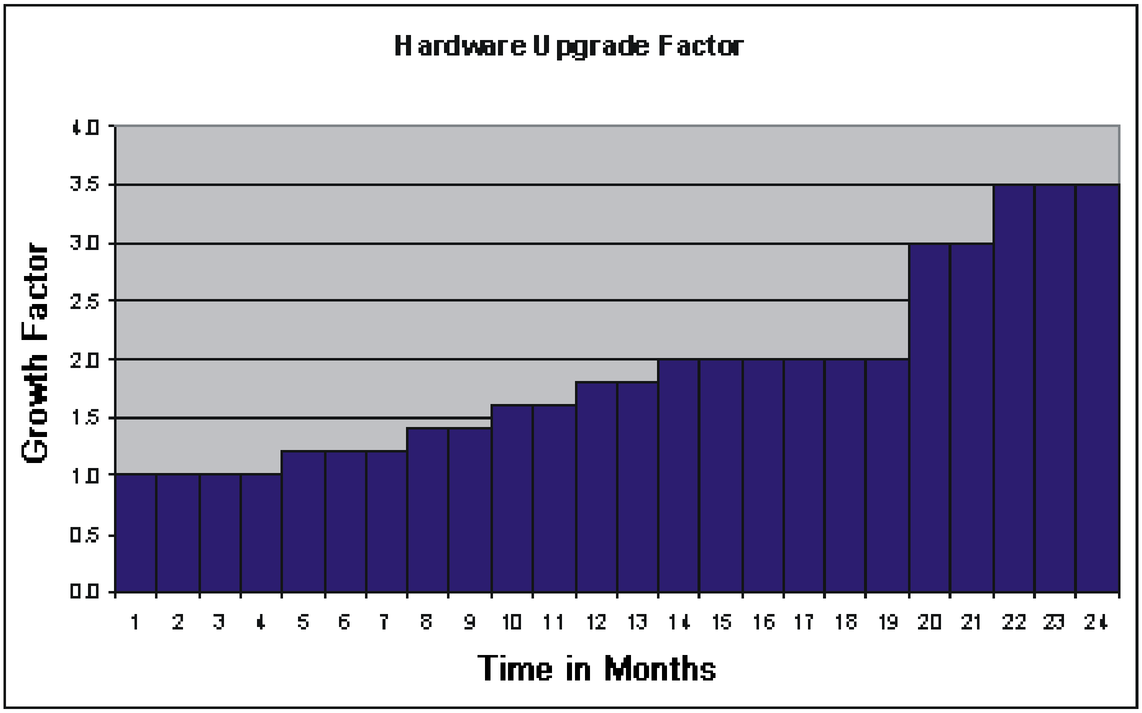


FIGURE 7 Hardware Upgrade Capacity Increase Factors

In Table 7 there is a column showing the CPU utilization in each month. This assumes a starting point which for this case is entered into the spreadsheet as 70% busy at month zero. The utilization is calculated by taking all the different growth factors into account, which is explained in the next section.

Combined Capacity Estimate

When all the load increasing factors are multiplied together, (i.e. the seasonal effect) such as the exponential workload growth and the marketing boost, a combined growth factor is the result. In this case, you can see that the final end point is about thirty times the starting point after two years, and that this boost mostly occurs at the end where seasonal and exponential growth combine.

Month	Workload	Marketing	Seasonal	Total Load
1	1.14	1.00	0.89	1.02
2	1.31	1.00	1.00	1.31
3	1.50	1.00	0.61	0.92
4	1.72	1.30	0.56	1.24
5	1.97	1.54	0.44	1.35
6	2.25	1.78	0.39	1.56
7	2.58	1.86	0.33	1.59
8	2.95	1.76	0.33	1.73
9	3.38	1.58	0.78	4.15
10	3.86	1.41	0.89	4.86
11	4.42	1.30	1.00	5.77
12	5.06	1.24	1.00	6.30
13	5.80	1.22	0.89	6.27
14	6.63	1.21	0.72	5.78
15	7.59	1.20	0.61	5.58
16	8.69	1.20	0.56	5.80
17	9.95	1.20	0.44	5.31
18	11.39	1.20	0.39	5.32
19	13.04	1.20	0.33	5.22
20	14.93	1.20	0.33	5.97
21	17.09	1.20	0.78	15.95
22	19.56	1.20	0.89	20.86
23	22.39	1.20	1.00	26.87
24	25.63	1.20	1.00	30.75

TABLE 7 Combined Load Calculation

On the other hand, the capability of the system is increased by hardware and software upgrades and application efficiency tuning. This is also tabulated by the spreadsheet.

Month	Hardware	CPU per Trans	Capability
1	1.00	1.00	1.43
2	1.00	1.00	1.43
3	1.00	0.80	1.79
4	1.00	0.80	1.79
5	1.20	0.90	1.90
6	1.20	0.90	1.90
7	1.20	0.90	1.90
8	1.40	0.70	2.86
9	1.40	0.70	2.86
10	1.60	0.55	4.16
11	1.60	0.55	4.16
12	1.80	0.55	4.68
13	1.80	0.55	4.68
14	2.00	0.50	5.71
15	2.00	0.50	5.71
16	2.00	0.50	5.71
17	2.00	0.30	9.52
18	2.00	0.30	9.52
19	2.00	0.30	9.52
20	3.00	0.20	21.43
21	3.00	0.20	21.43
22	3.50	0.20	25.00
23	3.50	0.15	33.33
24	3.50	0.15	33.33

TABLE 8 Combined Capability Calculation

The combination of efficiency and hardware capacity upgrades can be plotted on the same axis to see if the system is capable of sustaining the load. A national level of 70% utilization of capacity was also factored in, assuming that the system was 70% busy before the start.

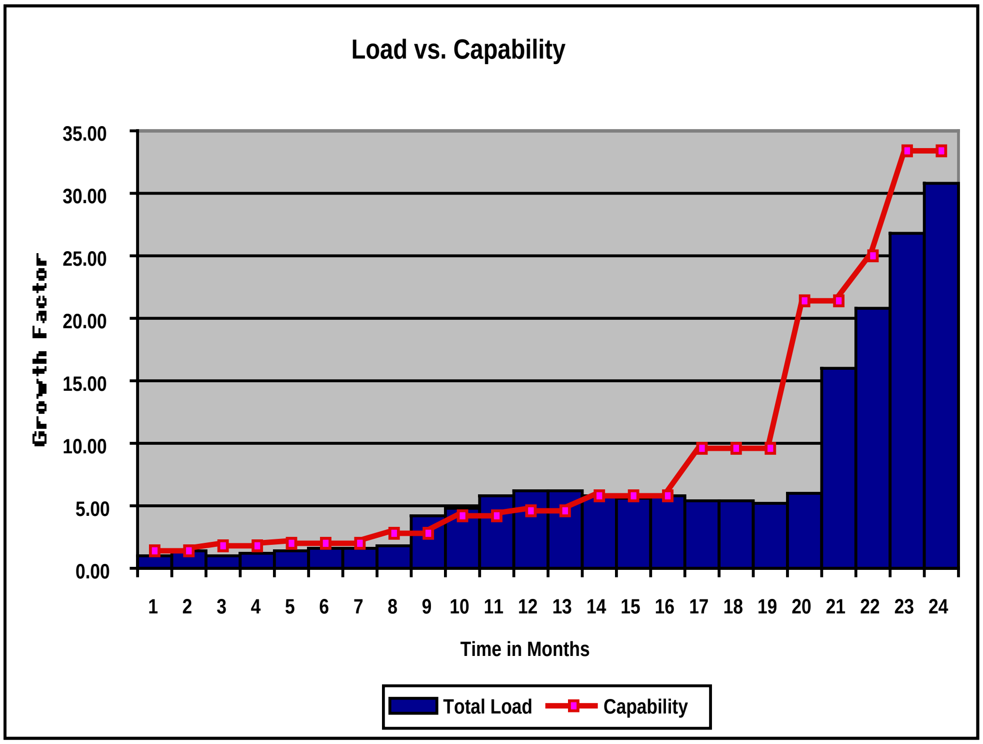


FIGURE 8 Combined Load Versus Capability

This combined plot shows that the combination of a marketing campaign and the seasonal boost overloads the server at peak times through the first christmas period. If this model is correct, something else needs to be done to sustain this future business scenario. Either a faster system could be installed earlier, or more efficiency could be squeezed out of the current system. However, another possibility is that the work itself could be split in some way. This is often possible using functional or geographical aspects of the workload to divide it up. If you were modelling the front end http servers, search engines, or application servers, the number of servers could be increased. It is much harder to split up a back end database.

Modifying the Model

The example used above has been carefully modified to give a capability that tracks the load level reasonably well. When you first enter your own data and estimates into the spreadsheet you will probably find that you have a very poor fit after the first few months. You need to spend time tweaking the parameters in the model and trying out alternative “what-if” scenarios until you are happy that your assumptions make sense.

With all this data, and some idea of the typical daily profile, you can predict the next few days in detail then monitor the real load level to see how close you get.

Summary

Capacity planning for some high growth E-Commerce sites can be described as driving a car at full throttle in the dark with no headlights, and every time the car wrecks you buy a faster one. I hope that the methods described in these two articles will entice you to do a small amount of work to illuminate the road in front of you a little.

Author’s Bio: Adrian Cockcroft

Adrian Cockcroft is a Senior Staff Engineer with the Enterprise Engineering group at Sun Microsystems, and the author of "Sun Performance And Tuning." Adrian is an accomplished performance specialist for Sun Microsystems and is recognized worldwide as an expert on the subject.