MULTIVARIATE IT CAPACITY MODELING

A real-world example of information technology delivering timely, actionable, intelligence

Ben Davies, Senior Consultant, Moviri INC – ben.davies@moviri.com
Andrea Vasco, Head of Operations, Moviri INC – andrea.vasco@moviri.com

1 EXECUTIVE SUMMARY

Wouldn’t it be great to be able to make a capacity model of an application that has multiple components and functions that are distributed over several devices, where those devices are shared with unrelated applications? This white paper examines this very real-world situation. The results achieved with actual customer data prove that not only is multivariate capacity modeling achievable, but it is relatively easily performed with COST software, tools and techniques.

In this real-life example, an IT Organization wants to model a two-fold increase in logins in the next quarter, resulting in the growth of two specific workloads, increasing usage by 350% and 450%, while the rest of the workloads remained essentially the same. The intuitive expectation was that the existing equipment would not accommodate the request, so the IT Organization needs not only to validate the expectation but also to produce options, driven by actual data and detailed analysis.

The outcome of using existing tools, available data, and skillful application of modeling and analysis techniques enables the IT Organization to create 4 different scenarios based on budget, risk, IT resource and business impact.

2 INTRODUCTION

A definition of multivariate analysis (MVA) is “*a statistical procedure for analysis of data involving more than one type of measurement or observation. It may also mean solving problems where more than one dependent variable is analyzed simultaneously with other variables*”. When applied to IT Capacity Modelling, MVAs are one of the most powerful analytical tools as they provide a framework capable of predicting the resource consumption (e.g. utilization of CPU) starting from multiple, independent inputs (e.g. different business processes). This technique is the most suited to support what-if scenarios for:

- **Increase application density**: select which applications or components can be hosted on the same infrastructure without compromising on performance
- **Evaluate the impact of business events** (increase in userbase, sales quotes, online check-ins, etc.) on shared infrastructure

The MVA described in this document has been created on behalf of a real IT Organization based on data collected from production environment. The resulting capacity models delivered timely, actionable intelligence, in the form of validation of the hypothesis that the current hardware would not sustain the expected additional load without violating SLAs. However, it also delivered several options that would address the business needs, each with a cost.

The starting point to build a multivariate capacity model is to identify the available, relevant, data sources such as APM tools, private cloud management tools, systems monitoring tools and metrics such as # of concurrent users, # of transaction by type, CPU utilization, memory utilization, etc. Then it is required to implement a capacity management database that can ingest the raw data from the identified data sources and output the identified metrics. Once the metrics are available it is possible to start performing various analysis and ultimately model few different scenarios.

The IT Organization we refer to in this paper is currently using AppDynamics as the APM solution, VMWare vCenter as the private cloud management tool, and BMC TrueSight Capacity Optimization (hereafter also referred to as “TSCO”) as the capacity management database. In this technology landscape, we decided that the native data collection provided by TrueSight Capacity Optimization could fulfill the role of the monitoring tool so that we could also leverage its capability to perform workload characterization. To enable the BMC tool to ingest data from

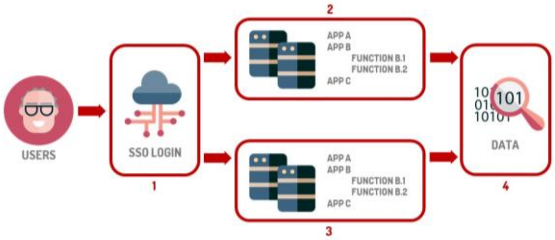
AppDynamics and VMWare we developed a custom integration for the APM solution and leveraged native integration for the private cloud tool.
This setup enables the IT Organization to effectively exploit the powerful analytical capabilities provided by TrueSight Capacity Optimization, and leverage the historical data provided by AppDynamics and VMware vCenter.

3 PROBLEM STATEMENT

The IT Organization runs multiple applications hosted on shared systems, and needs to understand how changes to a workload pattern will impact the shared systems that support them.

The business change affects an application and set of functions running on a set of shared servers, as follows:

- A new customer base will double the load on the SSO Login to application "B" (Box 1)
- The user base of application "B" function b.1 is expected to increase by 350% (Box 2 and 3)
- The user base of application "B" function b.2 is expected to increase by 450% (Box 2 and 3)
- All the other hosted applications and functions are not expected to change



4 SOLUTION

The IT Organization runs multiple applications hosted on shared systems, and needs to understand how changes to a workload pattern will impact the shared systems that support them.

4.1 FIRST STEP

Elicit the list the minimum required capabilities. Talking with the business, taking care to gather relevant information and practice active listening. Define success, inventory tools, inventory equipment, data sources, application structure, relevant business metrics, association tables and relevant business plans and opportunities.

4.2 SECOND STEP

Assess the available tools, infrastructure, data sources, business information, and opportunities to evaluate the overall success feasibility. Evaluating data sources and determining that they are not suitable to a given purpose is difficult and complicated, but the cost of introducing questionable data causes significant, and hard to detect, harm to the process. The more relevant tasks are:

- Validate all relevant applications are monitored by AppDynamics - supplying business metric data, system metrics, and association data, etc.
- Validate BMC TrueSight Capacity Optimization (TSCO from now on) is correctly configured and ingest required VMWare and OS monitoring data.
- Validate a custom ETL (Extract, Transform, and Load) operation can be designed and implemented to ingest AppDynamics data into TSCO.
- Validate actual CMDB coverage for the applications in scope and service model accuracy.

4.3 THIRD STEP

Provide guidance, expertly applied tools and techniques of the various tools, and leveraged the available data, distinctive feature sets and capabilities, to deliver actionable intelligence:

- Issue improvement recommendations for AppDynamics configuration, such as all relevant data must be collected using appropriate measurement units and made available to external tools.
- Secure the availability of the application owner to improve quality of CMDB data and service model accuracy.
- Design, implement and validate a custom integration between AppDynamics and TSCO, covering:
  - Performance metrics
  - Business transaction volumes
  - Service Models, dependencies and relationships
- Enrich service models by defining tags and additional grouping criteria, facilitating the model definition.

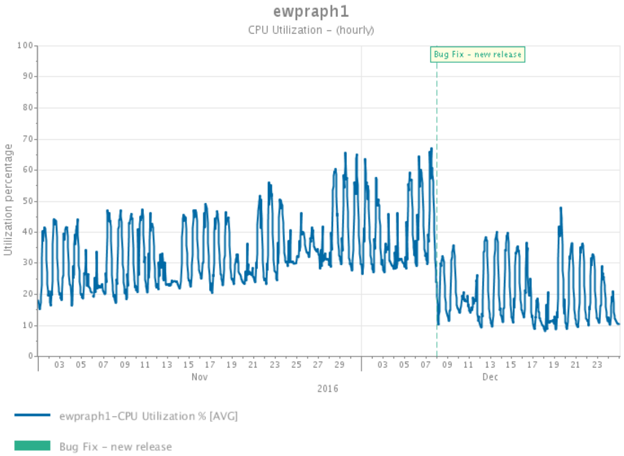
- Validate the correctness of analysis configuration by comparing the outcome of analysis to current data. Introducing errors in the analysis will adversely affect the accuracy of forecasts forcing to a spend a considerable amount of time in fixing it.
- Identify the most appropriate correlations between observed entities and metrics to provide options based on multiple KPIs.

5 CREATING THE CAPACITY MODEL IN TSCO

Creating the first capacity models revealed missing, mismatched and questionable data, which caused data sources to be interrogated, adjusted and ETL tasks modified. This was an iterative, interactive process with the business owners and equipment managers, that quickly yielded useful data, for useful capacity models and analysis.

5.1 IDENTIFYING A STEADY STATE

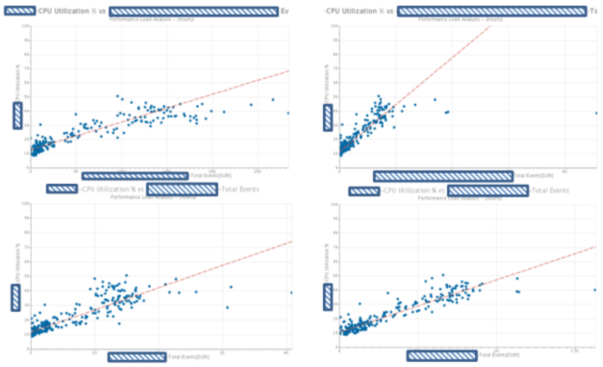
Identifying a steady state period that represented the 'normal' baseline was complicated by a known process 'bug' that impacted the resources of the subject systems. Without an identifiable steady or 'normal' baseline, model projections are significantly less reliable. Anomalies were effectively identified and eventually resolved, and a 'normal' state was produced.



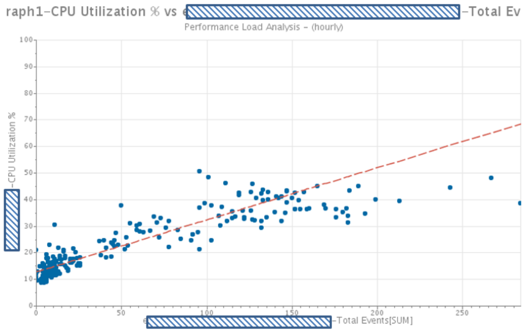
With multiple Performance vs. Time Analysis (PTA), Load vs. Time Analysis (LTA), Performance vs. Load Analysis (PLA), Performance vs. Performance Analysis (PPA), Load vs. Load Analysis (LLA) and Configuration data analysis, completed, compared and reviewed, a relatively complete understanding of the environment, application, and inter relationships became clear.

5.2 EVALUATING CORRELATIONS

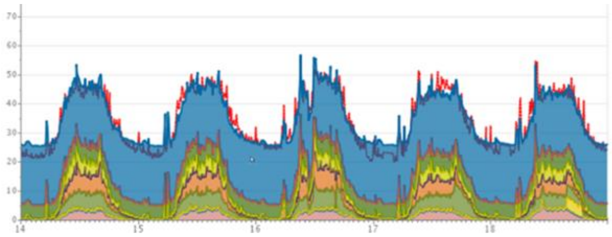
Here we have examples of Performance vs. Load Analysis. An expected outcome is a tight cluster following at a 30 to 60 degree angle (that implies an estimated service demand between 0.6 and 1.7, which is the expected value derived by previous analysis) and minimal outliers. Each workload was evaluated, anomalies examined, the potential of 'black swan' impacts, as well as unknown business marketing efforts, or seasonal spikes discussed. Some of the workloads were excluded from consideration, but they remained part of the model as not to lose their contributions to the total load.



The application workloads that were directly part of the application were given considerable scrutiny and found that there was a tight correlation between the workloads and CPU. It was determined that straight line regression was appropriate for these workloads, despite the slight leveling out after a critical amount of transactions per period.



With this understanding of inter relationships, simple models become progressively more complicated and the results analyzed and compared. Eventually, multivariate models are created, run and analyzed then compared against real data. The deviation from predicted vs reality indicate the reliability of the model. In our case, we achieved significant overlap, suggesting the model is accurate.

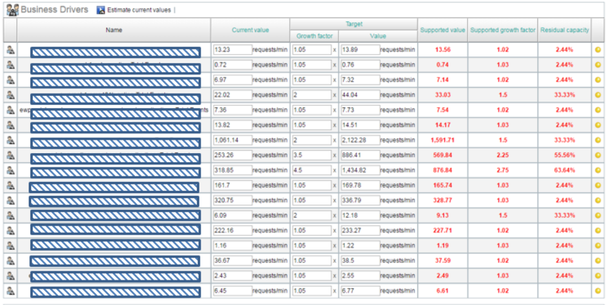


5.3 EVALUATING THE CAPACITY

To start, a base forecast was run for 90 days and did not violate thresholds. This suggested that without the expected business change, there is not an appreciable change to the environment for the near future. This is important to determine 'native growth rates', and to set a baseline.

A what if analysis was run increasing all workloads 10% and target workloads 500%. This violated CPU thresholds but not memory, storage or disk IO. This suggested that even with higher than expected growth the only resource driven above thresholds is CPU. These sorts of analysis are important to be sure that key resources are not overlooked.

The following what if analysis was run increasing all workloads 5% and target workloads by target percentages (logins 200%, function b.1 by 350% and function b.2 by 450%). This violated CPU thresholds but not memory, storage or disk IO. The multivariate capacity model predicted room for no more than ~5% of additional volume on the specified workloads. This analysis was the basis of **Option 1** or the '*do nothing*' option.

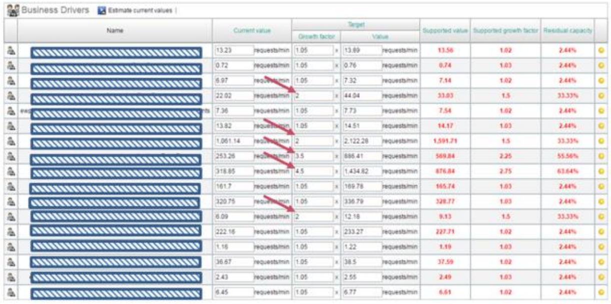

Business Drivers Estimate current values |

	Name	Current value	Target		Supported value	Supported growth factor	Residual capacity	
			Growth factor	Value				
	[redacted]	13.23 requests/min	1.05 x	13.89 requests/min	13.56	1.02	2.44%	●
	[redacted]	0.72 requests/min	1.05 x	0.76 requests/min	0.74	1.03	2.44%	●
	[redacted]	6.97 requests/min	1.05 x	7.32 requests/min	7.14	1.02	2.44%	●
	[redacted]	22.02 requests/min	2 x	44.04 requests/min	33.03	1.5	33.33%	●
	[redacted]	7.36 requests/min	1.05 x	7.73 requests/min	7.54	1.02	2.44%	●
	[redacted]	13.82 requests/min	1.05 x	14.51 requests/min	14.17	1.03	2.44%	●
	[redacted]	1,061.14 requests/min	2 x	2,122.28 requests/min	1,591.71	1.5	33.33%	●
	[redacted]	253.26 requests/min	3.5 x	886.41 requests/min	569.84	2.25	55.56%	●
	[redacted]	318.85 requests/min	4.5 x	1,434.82 requests/min	876.84	2.75	63.64%	●
	[redacted]	161.7 requests/min	1.05 x	169.78 requests/min	165.74	1.03	2.44%	●
	[redacted]	320.75 requests/min	1.05 x	336.79 requests/min	328.77	1.03	2.44%	●
	[redacted]	6.09 requests/min	2 x	12.18 requests/min	9.13	1.5	33.33%	●
	[redacted]	222.16 requests/min	1.05 x	233.27 requests/min	227.71	1.02	2.44%	●
	[redacted]	1.16 requests/min	1.05 x	1.22 requests/min	1.19	1.03	2.44%	●
	[redacted]	36.67 requests/min	1.05 x	38.5 requests/min	37.59	1.02	2.44%	●
	[redacted]	2.43 requests/min	1.05 x	2.55 requests/min	2.49	1.03	2.44%	●
	[redacted]	6.45 requests/min	1.05 x	6.77 requests/min	6.61	1.02	2.44%	●

The '*do nothing*' option should be thoroughly explored such that the key impacts and metrics can be identified and potential mitigations reviewed. This defines the cost of doing nothing, which is used to compare with the other options, and can galvanize the need to do something.

Several resolution options were identified and evaluated:

- Shift loads to different servers. This was the basis of **Option 2**
- Increments of 4 CPUs were added and results recorded (see picture below). These options became the basis of **Option 3**
- An underway project of a major software revision will add devices and reconfigure how parts of the application work. This project is a year plus from being ready, and involves some planned equipment expenditures. Speeding up this effort hardware wise, has significant financial planning impacts. This is the expected resolution before the capacity planning efforts, and is the basis of **Option 4**.


Business Drivers Estimate current values |

	Name	Current value	Target		Supported value	Supported growth factor	Residual capacity	
			Growth factor	Value				
	[redacted]	13.23 requests/min	1.05 x	13.89 requests/min	13.56	1.02	2.44%	●
	[redacted]	0.72 requests/min	1.05 x	0.76 requests/min	0.74	1.03	2.44%	●
	[redacted]	6.97 requests/min	1.05 x	7.32 requests/min	7.14	1.02	2.44%	●
	[redacted]	22.02 requests/min	2 x	44.04 requests/min	33.03	1.5	33.33%	●
	[redacted]	7.36 requests/min	1.05 x	7.73 requests/min	7.54	1.02	2.44%	●
	[redacted]	13.82 requests/min	1.05 x	14.51 requests/min	14.17	1.03	2.44%	●
	[redacted]	1,061.14 requests/min	2 x	2,122.28 requests/min	1,591.71	1.5	33.33%	●
	[redacted]	253.26 requests/min	3.5 x	886.41 requests/min	569.84	2.25	55.56%	●
	[redacted]	318.85 requests/min	4.5 x	1,434.82 requests/min	876.84	2.75	63.64%	●
	[redacted]	161.7 requests/min	1.05 x	169.78 requests/min	165.74	1.03	2.44%	●
	[redacted]	320.75 requests/min	1.05 x	336.79 requests/min	328.77	1.03	2.44%	●
	[redacted]	6.09 requests/min	2 x	12.18 requests/min	9.13	1.5	33.33%	●
	[redacted]	222.16 requests/min	1.05 x	233.27 requests/min	227.71	1.02	2.44%	●
	[redacted]	1.16 requests/min	1.05 x	1.22 requests/min	1.19	1.03	2.44%	●
	[redacted]	36.67 requests/min	1.05 x	38.5 requests/min	37.59	1.02	2.44%	●
	[redacted]	2.43 requests/min	1.05 x	2.55 requests/min	2.49	1.03	2.44%	●
	[redacted]	6.45 requests/min	1.05 x	6.77 requests/min	6.61	1.02	2.44%	●

6 THE OBSERVATIONS AND PROPOSED SOLUTION.

The actionable intelligence that resulted from this investigation was:

- Two devices were modeled to violate thresholds, and all modeled the same way. The capacity models were run on one device and would be applied to the impacted devices. Several devices made up the system and all were reviewed, but only two were significant.
- The constrained resource was CPU. Memory, storage and disk IO were evaluated but did not exceed threshold.

These options were developed with input from the business and estimates of the cost in money, effort, risk, and business impact were created. **This allowed for a business decision that is based not only on money or available resource, but on impacts to the business**. It was assumed that Option 4 would be the path forward (more on that in option 4 description) but these observations and proposed solutions provided alternatives, costs and impacts that could be evaluated and compared.

6.1 OPTION 1 - KEEP IT AS IT IS.

Is the 'do nothing' option. This had a high likelihood to result in a CPU bound device that would negatively impact response time during peak use and would negatively impact several unrelated applications, as well as the target application. Note that it is helpful to explore the do nothing option first and in detail, as this crystallizes what the impacts and costs of doing nothing are. As the likelihood is large, and the negative impact is large, and the cost to customer satisfaction and reputation is large, people were motivated to 'do something', and had a cost to compare the other options against.

6.2 OPTION 2 - WORKLOAD MIGRATION

Is to shift other loads to different devices. This had little financial impact as the equipment was on site but not in the 'proper place' on the network. Implementing this required moving existing services to other devices, physically moving the devices to the proper network segments, and configuring traffic. This option has significant configuration changes to existing applications, firewalls, load balancers, networking and redirecting traffic of other applications to these devices. While not monetarily expensive, the manpower effort, and operation risks, are significant.

6.3 OPTION 3 - SCALE VERTICALLY

Is to add CPU to the existing named devices (a virtual device hosted on large physical hardware). This had some financial impact as some of the (unrelated) software licensing is CPU driven, but this has less configuration changes, therefore less operational risk than Option 2. However, this removed much of the reserve capacity of the physical device.

6.4 OPTION 4 - SCALE HORIZONTALLY

Is to speed up an underway project of a major software revision that will add devices and reconfigure how parts of the application work. This project is a year plus from being ready, and involves some planned equipment expenditures. Speeding up this effort hardware wise, has significant financial planning impacts, complicates the eventual rollout of the upgrade, and all of the manpower effort and configuration risk of Option 2.

It is interesting to note that this exercise allowed the customer to avoid speeding up option 4 which is work in the pipeline, and was the assumed to be the answer before the capacity planning exercise. For the record, it was assumed that there was not enough reserve CPU to handle the increase based on simple capacity estimates of tripling the entire load on the existing servers. The ability to do multivariate analysis showed that increasing specific loads (not all) allowed the business to choose the less impactful choice with confidence.

7 CONCLUSION

The result was that skillful application of tools and techniques within the Capacity Optimization tool allowed additional value from existing data sources. Avoided unnecessarily activating a future plan to accommodate a short-term business opportunity, while accurately modeling a complicated impact to existing applications in a complicated environment. This value was delivered quickly, with data driven deaccessioning, in an audible, repeatable, way, without programmatic changes to any of the systems or applications involved, using built in features.

This result, this actionable intelligence, is achieved with smart use of the tools and data available along with detailed conversations with the business. Exactly what the experts say is needed to run IT like a business.