PROGNOSTIC CAPACITY MANAGEMENT FROM AN IT SERVICE MANAGEMENT PERSPECTIVE

Thomas Jirku
University of Applied Sciences "Technikum" Vienna, Austria

Peter Reichl
Telecommunications Research Center Vienna (ftw.), Vienna, Austria

Keywords: IT Service Management, ITIL, Capacity Management.

Abstract: Maintaining the balance between keeping the IT continuously running and at the same time enhancing the quality of the services and responding with increasing agility towards changing business needs under the usual budget constraints is one of the core challenges for IT service management. Over the last years, ITIL has become the de facto standard best practice recommendation for this purpose. Among the 10 different ITIL disciplines, capacity management is one of the most important ones. However, the ITIL perspective is restricted towards describing the current state of an IT system. In this paper, we propose to go a step ahead and turn capacity management into a proactive discipline for managing IT infrastructure. To this end, we discuss requirements on a platform-independent capacity data base and present a case study which shows how to simplify its performance evaluation through an approximation approach. Finally, we draw a couple of practical conclusions and describe further steps towards a capacity management which allows to deal proactively with potential problems and bottlenecks.

1 INTRODUCTION

Due to increasing cost pressure and complexity, it has become more and more important to proactively manage the IT service landscape in order to identify and potentially avoid expensive bottlenecks and/or corresponding outages of IT operation in advance. To this end, the complexity of enterprise processes is reduced to create a joint standard for an integrated, simplified and unified IT service management. In this context, the establishment of a common database, the so-called Configuration Management Database (CMDB), allows all users and processes to have access to identical information, and thus has become one of the central concepts within the framework of the IT Infrastructure Library (ITIL).

Among the different ITIL disciplines, capacity management is of specific importance, as an efficient and comprehensive capacity management can lead to early recognizing and effectively counteracting future bottlenecks in the IT. This allows to use scarce budget resources more efficiently and to increase the satisfaction of the end customers also on a long-term timescale.

This paper discusses several innovative aspects of ITIL capacity management. After a brief overview on fundamental ITIL concepts and capacity management in general, we first deal with the requirements on the CMDB and propose a concept for a system- and platform-independent capacity management database. We then describe a characteristic reference service comprising two tiers of components, and derive requirements on corresponding workload models. We then use this service for a case study discussing capacity-related performance evaluation. Having indicated the standard approach, we then demonstrate how to simplify this quite complex task by using a queueing-theoretic approximation approach based on Little's Law. In the final section, we summarize our work, before we draw some conclusions and point out several directions for further work.

Jirku T. and Reichl P. (2008).
PROGNOSTIC CAPACITY MANAGEMENT FROM AN IT SERVICE MANAGEMENT PERSPECTIVE.
In *Proceedings of the Tenth International Conference on Enterprise Information Systems - ISAS*, pages 320-325
DOI: 10.5220/0001710603200325

2 A BRIEF HISTORY OF ITIL

The IT Infrastructure Library (ITIL) has been developed almost 30 years ago as an initiative of the British Central Computing and Telecommunications Agency (CCTA, today Office of Government Commerce, OGC). Actually, work focuses on the final version 3 (ITIL Refresh) whose framework has been published on June 1, 2007 (Poizat 2007).

ITIL has been compiled in order to facilitate the planning, monitoring and controlling of high-quality IT services, and over the years has become the de facto standard best practice in this field. Thus, today ITIL is the only one comprehensive non-proprietory process library focusing on provision and compliance of IT services according to a process-oriented model.

The fundamental idea of ITIL is related to a central and jointly used data base, the *Configuration Management Database* (CMDB) which contains all relevant configuration information of the so-called *Configuration Items* (CI) of the system (like software, hard disks etc.). The CMDB consolidates information which otherwise would be spread between e.g. problem, change and process data bases, and thus allows an efficient and transparent access to these data.

Based on functionality, two main parts of ITIL are distinguished. *Service Support* deals with IT services on an operational level, including service desk, incident management, problem management, configuration management, change management and release management. On the other hand, *Service Delivery* is concerned with planning and operating processes for a professional provision of IT services. Corresponding issues include service level management, financial management, capacity management, continuity management and availability management. Among these, we will concentrate on capacity management as a key discipline for the resolution of incidents and pre-identification of capacity-related problems.

Note that the detailed understanding of business requirements and corporate processes is an essential prerequisite for capacity management to be capable of dealing with current and future developments both in economics and technology. Capacity management processes target the complete hardware infrastructure, peripherical systems, the complete software infrastructure and to some extent even human resources. In this framework, capacity management provides information about the current (and ideally also future) resource usage of individual components and services in order to enable well-founded and fact-based decisions for the enterprises.

3 REQUIREMENTS FOR A SYSTEM - AND PLATFORM-INDEPENDENT CAPACITY DATABASE

It has already been mentioned that the capacity database is among the most important ideas to be found at the heart of ITIL. In order to comply with the model of a comprehensive and system- and platform-independent capacity database which is suited to provide short- and long-term predictions of IT utilization and performance of capacity, the following requirements have to be met:

- Java-like independence of lower layer systems and platforms;
- High scalability and availability of the data base with respect to size and storage capacity (including concepts of federated data base architectures and data staging);
- Compliance with existing standards for data import and export (in the sense of a data warehouse);
- Free choice of Key Performance Indicators (KPI) of the performance models;
- Conducting "what if"-scenarios through directed change of certain data sets for simulation of developments and changes within the enterprise;
- Rapid identification of "root causes" through correlation with other incidents and problems to allow precise predictions of future developments and potential problems;

Note that most widely used commercially available data bases, like Microsoft SQL, ORACLE, DB/2 and Ingres, heavily depend on the operating system underneath. As a very prominent example consider Microsoft SQL, where the SQL database server may only be installed on the windows operating system. Therefore, a data base system would be desirable which may be installed regardless of the operating system and then runs without constraints.

There is already a clear and well established analogue to this idea within the area of programming languages, i.e. Java with its notorious virtual machine concept. In a similar way, the functionality of a platform-independent data base could be specified according to Figure 1.

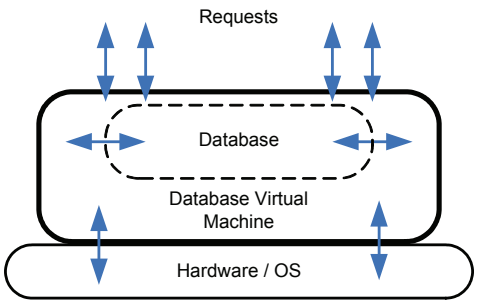

Figure 1: Concept of a Database Virtual Machine.

For further details on the Database Virtual Machine we refer to (Jirku 2008).

4 PROGNOSTIC CAPACITY MANAGEMENT – A CASE STUDY SCENARIO

Having discussed properties and requirements of the central capacity data base, we now introduce prognostic performance evaluation, focusing on the following reference service as an interesting initial case study.

4.1 Definition of a Reference Service

Following the current strong trend towards web applications in enterprises which more and more replace traditional mainframe applications and/or classical client/server applications, we consider the following rather generic scenario for our subsequent performance evaluation:

- An end user stays at the edge of the network, together with a router and several client PCs. The customer router connects to the WAN, the corporate router relates the WAN and the corporate LAN of the central site. The client and the corporate networks are a 100MBit Ethernet network.
- The end users access a portal from their client PCs on a web server in the central site, in order to receive data. As a protocol they use http 1.1 over TCP/IP.
- The users demand data from the web server to generate a report. To this end, texts and pictures of different size are transmitted.
- The web server prepares the report requested and makes it accessible to the user. Finally, the report is shown on the client PC screen.

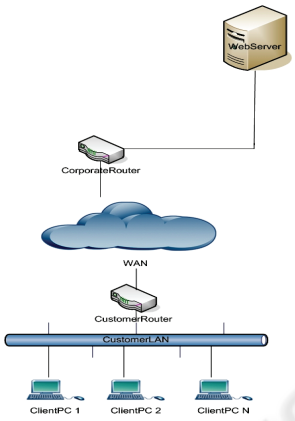

Figure 2: Architecture of the Reference Process.

Figure 2 illustrates this “two tier models”, based on the information flow from client PC to the web server and back. Note that the clients involved can be classified as “thin” and “fat” clients, where the thin clients (a.k.a. network computers) lack of computing power as well as local hard disks and thus depend on a server and cannot operate on their own. In contrast, fat clients are computer systems disposing of significant computing power and permanent storage of their own.

Additionally, we include the possibility of caching sought-after data e.g. on the system of the end user or in between, in order to prevent resource wasting because the client has to access the server(s) for every individual request. For further details on cacheing strategies we refer to (Zeng 2004).

4.2 Workload Concepts

In order to achieve reliable capacity predictions and to be able to calculate and understand the behaviour of IT services under different load scenarios, we need precise workload models, i.e. the mapping of real load models characterised e.g. by increasing user demands. This includes not only metrics like CPU or hard disk usage, but also busy hours, public holidays or marketing campaigns which may influence the user behaviour and thus the workload significantly.

Summarizing, workload concepts have to take the following requirements and steps into account:

- The location of the equipment measuring response and transaction times has to be specified;
- The relevant parameters for prognosis have to be specified;

- The system has to be monitored for a certain time *t* to collect historical data as a basis for modelling;
- Aggregation of data for reducing the amount of data to a necessary minimum;
- Compilation of a workload model corresponding to reality;
- Plausibility-check of all parameters and metrics characteristic for the model to guarantee precision.

The load e.g. of a client-server has to be described from different perspectives. With respect to the reference process described in section 4.1, three different perspectives can be derived: The *business model* refers to metrics like the number of available items which can be ordered in a web store, or bills per customer generated by an ordering system. On the functional layer, parameters which describe how requests are processed by applications are described, meaning that how many transaction can be processed by the billing application itself. Finally, the resource-oriented lowest layer includes resources whose usage may lead to long transaction times or high CPU usage. Figure 3 sketches the resulting layered workload model. For a more detailed explanation on different workload models we refer to (Menascè 1994).

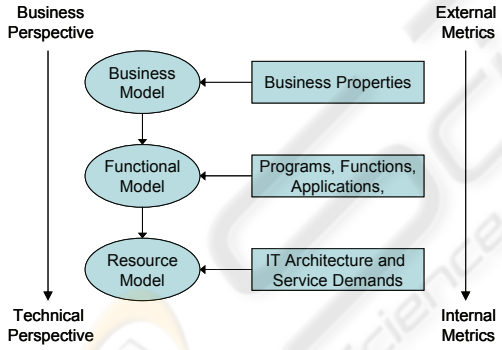

Figure 3: Layered Workload Model.

5 HOLISTIC PERFORMANCE ANALYSIS

In a strict sense, ITIL and the general concept of IT services only allow evidence about the current state of IT services. As an IT service usually consists of several individual components communicating with and depending on each other (see section 4.1), we consider a holistic perspective on the performance of the IT service as appropriate, which is of course composed of the individual components and their performance.

5.1 Parametrization

Based on the reference scenario described in the previous chapter, we now demonstrate how to calculate the performance, based on the following couple of assumptions: Users in the customer LAN access a portal page at the web server. The clients' network as well as the network of the central office is a 100MBit Ethernet network, the WAN connection between both networks is an FDDI network with 2MBit bandwidth. The documents requested from the web server are assumed to have two different sizes, i.e. 10kByte and 100kByte. During one hour, 1000 requests arrive (i.e. 0.27 requests per sec), 20% of them asking for large documents. The request itself sent from client to server has length 300 Byte. Moreover we assume that the block length on the web server's hard disk is 2,048 kByte, with a mean access time of 9ms, mean latency of 4.17ms and mean transfer rate of 20MB/s (note that all values in this paragraph correspond to actual specifications or other real data). Requests and responses have to pass two routers each, which currently are able to process 400,000 packets per sec. Finally, our observation shows that the web server needs 5ms CPU time on average per request and is able to process 10 requests in parallel.

5.2 Standard Performance Evaluation

A straightforward calculation of the time which client requests and web server responses need to cross the network, taking the assumed technologies into account, yields the following results: the 300 Byte request sent by the client including IP headers and Ethernet overhead requires a minimum of 6.529ms before arriving at the web server. The network time of the web server's responses are different for the two types of responses: the short response requires a total of 0.024ms, whereas long responses need 0.069ms. Note that the main proportion of this response time is caused in the WAN between customer and corporate router, as is demonstrated in Figure 4. Hence, the WAN between the central office and the branches can be identified as the bottleneck of the system. Further analysis shows that increasing the FDDI bandwidth to a value of 8MBit/s or more would resolve this problem.

As far as the service time caused by the client request during access to the disk of the web server is concerned, we assume that data on the hard disk are

split into blocks of 2048 Bytes each. Furthermore, the specification of an off-the-shelf IDE (Integrated Drive Electronics) hard disk reveals that the mean access time is 9ms, the mean rotation latency 4.17ms and the data transmission rate equals 20MB/s at a rotational speed of 7,200 rotation/min; the controller time is 0.1ms. Based on these numbers, a straightforward calculation shows that the time required for reading the data blocks from the hard disk equals 13,3 ms. In reality, we often find hard disks of the type SCSI (Small Computer System Interface) operating at a rotational speed of 15,000 rotations per minute yielding an average access time of only 3ms.

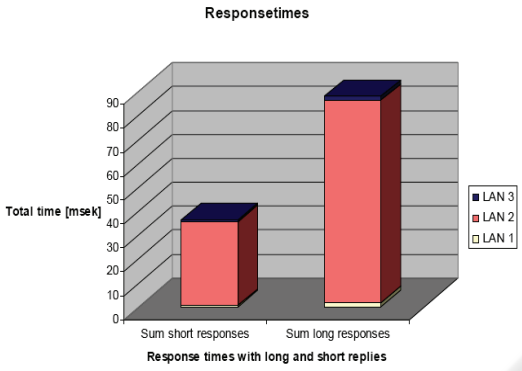

Figure 4: Response Times. Note that LAN1 refers to the Customer LAN, LAN3 to the Corporate LAN and LAN2 to the WAN in between.

Summing up these values, we end up with a total service time of 108.46ms for short responses and 806.76ms for long responses. Under the initially assumed distribution of 80% vs. 20%, this finally yields an average expected total service time of 457.61ms. Figure 5 demonstrates how this time is distributed between disk access, network, routers and CPU. For more details on the calculation we refer to (Jirku 2008).

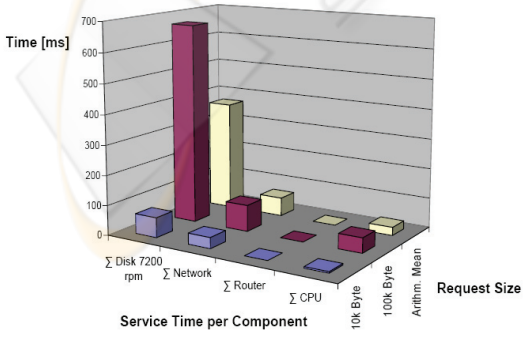

Figure 5: Total Service Times for Request Sizes of 10kByte, 100kByte and Average Size (80% vs. 20%).

5.3 Approximation with Little's Law

The exact calculation of the total service time as sketched in section 5.1 may become very complicated and time-consuming in the case of large real-life data sets. Therefore, we propose to use Little's Law for simplifying this process:

$$N = \lambda \cdot R \tag{5.1}$$

for an average number of requests N, average arrival rate λ and average processing time R.

According to (Haverkort 1998), Little's Law can be applied to a vast variety of queueing theory scenarios. We know that the web server handles an average number of 1,000 requests per hour, and that the server is processing one request at a time. Therefore, in this case (5.1) results in

$$R_W = \frac{N_W}{\lambda_W} = \frac{1}{10000/3600} = 350ms \tag{5.2}$$

as web server response time.

Both routers in our example have an average throughput of λ = 400,000 packets per second and may process 2 packets in parallel, therefore we end up with a total router time for the 2 routers in our system of

$$R_R = 2 \cdot \frac{N_R}{\lambda_R} = 2 \cdot \frac{2}{400000} = 1\mu s \tag{5.3}$$

For the network, we can assume an arithmetic average message length of 55kByte and thus end up with

$$R_N = \frac{N_N}{\lambda_N} = \frac{55000}{400000} = 140ms \tag{5.4}$$

Finally, the corresponding CPU time is

$$R_C = \frac{N_C}{\lambda_C} = \frac{10}{0.27} = 37ms \tag{5.5}$$

Summing up these values, we derive the average service time for a mean request size of 55kByte as

$$R_{tot} = R_W + R_R + R_N + R_C = 527ms \tag{5.6}$$

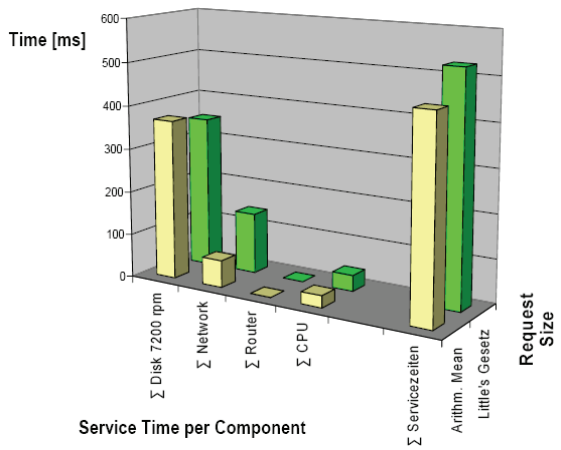

Figure 6: Total Service Times: Average vs Little's Law.

On the other hand, averaging the precise values for short and long requests as discussed in section 5.1 yields a mean service time of approx. 457ms which is reasonably close to the approximation value in (5.6). Therefore, we may safely conclude that using Little's Law for capacity calculations can lead to very good results at a significantly lower complexity than conventional calculation.

Figure 6 summarizes both approaches and demonstrates that the approximation also depicts the distribution of the total service time correctly. For further details we refer to (Jain 1991) and (Jirku 2008).

6 CONCLUSIONS AND FUTURE WORK

This paper discusses how to integrate the idea of proactive capacity management into the current ITIL framework. We have presented a list of requirements both concerning the CMDB and the workload modelling which have been derived from our long-time practical experience with issues of IT Service Management in various companies. Our second contribution deals with efficiency of the required performance analysis, where we illustrate that "Little's Law", which is well known from queueing theory, can lead to a significant reduction of computational complexity while still leading to sufficiently precise results and predictions.

Knowing the capacity not only of single IT-infrastructure related items but also external factors like promoting new products, which may result in more requests to ones IT resources, became more and more vital to the role of today's CIO (Schubert 2004). Being able to predict the capacity of networks and/or systems, and relate different measures, errors and faults to IT services their overall performance will become a key part in IT capacity planning. While the operational part of capacity planning will remain to be quite computationally intensive, it even will be a bigger challenge to translate the business needs to measurable performance indicators, like if the IT infrastructure is able to handle increased user access due to a promotional campaign Therefore, our main focus now is on algorithms and strategies that relate the user experience of a slow IT service to a poorly performing router or a slow hard disk response because of fragmented disks, which have to be developed and continuously improved to reflect the real world. However, we believe that it is worth this effort due to the huge resource saving potential which ITIL-compliant prognostic capacity management will provide especially in the areas of system virtualization and "green IT".

ACKNOWLEDGEMENTS

The authors would like to thank Thomas Sommer and Christian Kaufmann for their continuous and valuable support. Part of this work has been funded in the framework of the Austrian Government's Kplus/COMET competence center program.

REFERENCES

Jirku, T., 2008: Prognostic Capacity Management from an IT Service Management Perspective. Master thesis, University of Applied Sciences "Technikum" Vienna, Austria, 2008.

Poizat, C., 2007: The IT Infrastructure Library [online]. http://itil.technorealism.org/index.php

D.A. Menascè, V.A.F. Almeida, and L.W. Dowdy, Capacity Planning and Performance Modelling: From Mainframes to Client-Server Systems, Prentice Hall, Upper Saddle River, New Jersey, 1994.

Jain, Raj, 1991: The Art of Computer Systems Performance Analysis, New York, 1991.

D. Zeng; Fei-Yue Wang; Mingkuan Liu, 2004: Efficient web content delivery using proxy caching techniques, Manage. Inf. Syst. Dept., Univ. of Arizona, Tucson, AZ, USA, 2004.

Schubert, Karl D., 2004. CIO Survival Guide: The Roles and Responsibilities of the Chief Information Officer, New Jersey, USA, 2004.