1 Introduction

Allocation of IT infrastructure costs is an increasingly important topic in corporate IT departments. According to a study by Forrester [Forr06] computer hardware (21%) together with Networking and Communications hardware (14%) account for more than one third of current IT budgets. Typically, these infrastructure costs (e.g., servers and networking infrastructure) are allocated to business units based on simplified cost allocation keys [Sysk02; GaMa05, p. 136]. Comprehensive IT Asset Management and chargeback solutions have found limited use as of yet [GaJK05]. Usage-based accounting and pricing should lead to higher transparency and has recently become a major topic for CIOs. Nowadays enterprise software architectures are mostly designed as multi-tier client-server applications. In such environments the determination of usage-based allocation keys is very challenging [Bert01]. If the hardware is dedicated to specific customers, for instance business units, those costs can be treated as direct costs. The actual resource consumption (e.g., CPU time or number of I/O operations) is not a relevant cost driver and can be ignored. However, more and more IT infrastructure is nowadays shared among multiple applications and business units. Typical examples include database servers, application servers or virtualized servers (e.g., using VMware [Vmwa] or Xen [UCCL]). In these cases the resource consumption of the installed applications and the workload generated by the customers is a significant cost driver. An average application server in an industrial data center (e.g., 4 CPU / 16 GB memory) can easily host dozens of light-weight applications, whereas in other workload scenarios, the same server may be fully utilized by one or two applications. If the costs for such a server are treated as indirect costs and are apportioned via flat rates or fixed percentages, the IT Management, as well as the concerned business units, has only few possibilities for cost controlling and planning. Furthermore, the server is usually not offered as standalone product, but as part of a larger IT system. A request in a 3-tier database application, for example, comprises a web...
server, an application server, and a database server, which are all typically used by different applications.

A cost accounting approach, which ignores the resource consumption, may lead to multiple free-rider problems. For example, application owners do not consider the resource requirements when selecting off-the-shelf software. Also, the owners of “light-weight” applications might have to bear a very high share of the cost for a particular application or database server, which in turn makes it more difficult to finance these applications. Obviously, a usage-based model, where IT infrastructure costs can easily be allocated to application owners or even to users directly increases cost transparency and would have a number of advantages. A technical possibility would be to determine consumption-based cost allocation keys by detailed monitoring and metering of each service request. This would require assigning a unique ID for each user to each database request and each thread running on an application server in order to determine exactly how much of the resources a service customer has consumed. It would force the adaptation of the entire IT infrastructure, cause a huge monitoring and metering overhead, and is typically not viable (irrespective of the fact that a business user probably would not accept technical accounting metrics as the CPU times of different servers).

In this paper, we propose a method to determine usage-based cost allocation keys for customer-oriented services based on their estimated resource consumption. Deriving such an estimator, however, is a non-trivial task.

First of all, the estimator should be unbiased, in the sense that on average it should not over- or underestimate the true resource consumption.

Second, the estimation should be applicable to various IT infrastructures (i.e., different hardware, operating systems and applications), without a need to change the respective systems.

Third, the estimation should cause little extra work and integrate well with existing IT service management processes. While the first requirement is essential for the cost allocation key to achieve incentive compatibility, the second and third requirements target the viability of the approach. Clearly, direct apportioning of IT infrastructure costs could easily outweigh the benefits of a usage-based cost allocation key.

Based on a series of load tests, we derive for every service a so-called resource profile as estimator for its true resource consumption. We consider CPU time, storage I/O, and network traffic, which typically are the scarce resources, and consequently cost drivers and parameters for new investment decisions. This profile multiplied by the number of service invocations in an accounting period can then be the basis for a usage-based cost-allocation. The actual charges for business units might of course also be influenced by other non-cost-based factors, such as management incentives to use certain applications.

A main question is how well such an estimator meets the real resource consumption, given different workloads and different usage behavior in today’s complex IT infrastructures. Only, if the estimates are of high accuracy, business units will accept a respective cost allocation key. We use Queuing Networks to validate the estimated CPU times and predict server utilization for different workloads. These predictions are compared and tested against the parameters measured in load tests with respective workloads.

Finally, we conducted several experiments with J2EE applications in a distributed client/server infrastructure consisting of Unix, Linux, and Windows servers in a data center of the BMW Group and achieved very promising results even in a very heterogeneous environment with multiple software modules, operating systems, and hardware infrastructures. The estimation procedure could be integrated with the IT service management processes at the central IT division of the BMW Group with little extra effort and is applicable to different types of IT infrastructure including server capacity, storage I/O, and network bandwidth.

The remainder of the paper is structured as follows. Section 2 briefly discusses current approaches. Section 3 describes the concept of allocating infrastructure costs via resource profiles. Sections 4 and 5 present methods for estimating resource profiles for IT services and describe how we use Queuing Network Models for validation and capacity planning. The concept is illustrated by experiments with the example application Java Pet Store from Sun Microsystems [SuMi]. Section 6 compares the approach with related work. The paper concludes with a short summary in section 7.

2 Survey of Literature and Current Practice

Cost accounting for IT infrastructures falls under the realm of IT (Infrastructure) Costing (see for instance [Karg99; KrBR00; GaMa05]) or, from the ITIL perspective [OoCC01], IT Financial Management. Several concepts exist (e.g., cost-center accounting, process costing). The choice of an appropriate approach is mainly dependent on management requirements and the organizational structure of the IT unit (e.g., service center, cost center, profit center). Overall, cost accounting can be divided into three major steps (figure 1) [Gart03].

Cost identification is an organizational/accounting issue. It focuses on making IT costs visible and assigning them to accounting objects, such as the provided IT services. Cost allocation distributes the costs to the business units, enables the assessment of their financial performance and improves forecasting and decision making. It deals mostly with technical aspects, such as measuring usage and identifying cost allocation keys for the different IT services. Finally, cost recovery describes the process