CAPPLES – A Capacity Planning and Performance Analysis Method for the Migration of Legacy Systems*

Paulo Pinheiro da Silva[1], Alberto H. F. Laender[2], Rodolfo S. F. Resende[2], and Paulo B. Golgher[2]

[1] Department of Computer Science
University of Manchester
Oxford Road, Manchester M13 9PL, UK.
pinherp@cs.man.ac.uk

[2] Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
31270-970, Belo Horizonte MG, Brazil.
{laender,rodolfo,golgher}@dcc.ufmg.br

Abstract. Many organizations have a number of mission-critical systems that are out-of-date, but that are essential to their activities and cannot be discontinued. This problem is known as the *Legacy System Dilemma*, and it is usually solved by the migration of the existing systems to a completely new environment. Although there are many strategies and tools to perform this migration, no methods are available for evaluating the performance of the new system before its migration has been completed. This paper presents CAPPLES, a capacity planning and performance analysis method for the migration of legacy systems. A real case study is presented where CAPPLES was successfully applied to predict the behaviour of a new version of a mission-critical legacy system. Details of how to use CAPPLES, such as the characterisation of the synthetic workload and the simulation of the new system, are also provided.

1 Introduction

Many organisations have a number of mission-critical systems that are out-of-date and very difficult to maintain [2, 6, 8], but that cannot be discontinued. These systems, known as *Legacy Systems*, have became in evidence these days particularly due to the so-called Year 2000 bug [12]. There are many ways of minimizing the problems related to legacy systems. However, the migration of these systems to a new environment is usually the most effective approach to solving such problems. The new systems that originate from this migration are known as *target systems*. Several strategies [3, 4, 6, 25] and tools [10, 20] have been proposed to help the migration of legacy systems. For instance, reverse engineering [9, 21, 23] is one such a strategy.

* This work was supported by COPASA-MG under cooperative agreement 3025.

On the other hand, during the life cycle of a system it is usual to perform capacity planning and performance analysis studies not only to evaluate the system's environment but also to plan its operation [13, 26]. This type of study would also be very useful when planning the migration of a legacy system. However, traditional capacity planning and performance analysis methods [11, 16, 19] cannot cope with some problems concerning the migration of legacy systems. The first problem is that two systems must be considered: the legacy system itself and the target system. The second problem is that legacy systems are usually very large (sometimes with more than one million lines of code) and their target systems tend to be even larger, which makes the performance analysis study very complex. Moreover, the time required to perform the study is in general extremely long, whereas the time available to understand the whole migration context is very short. Therefore, any effort to understand the environment of both systems, as required by the traditional methods, should spent more time than is available. A third problem is that the workload characterisation must be based on information from both systems. The traditional methods do not explain how such workload characterisation should be carried out.

Thus, although there are several strategies and tools available to migrate legacy systems, there are no methods to predict the behaviour of the target systems when legacy systems are shutdown. Someone can argue that adopting an incremental migration strategy, as proposed by Brodie and Stonebraker [6], an organisation can reduce the migration risk to a minimum. However, we recall that incremental migration is only possible for decomposable systems (as defined in [6]), and that minimal risk is not the same as no-risk.

This paper introduces CAPPLES, a CApacity Planning and Performance analysis method for the migration of LEgacy Systems. The method was originated from a specific need at COPASA-MG, the Water and Sewage Company of the State of Minas Gerais, Brazil, whose development team had to evaluate the performance of a target system before it became operational. Therefore, this paper presents a real case study where CAPPLES was successfully applied to predict the behaviour of a non-operational target system of a mission-critical and operational legacy system.

The remainder of the paper is organised as follows. Section 2 defines some terms and presents the assumptions considered in this work. Section 3 presents CAPPLES. Section 4 describes SICOM, the COPASA-MG's system which is the subject of this study. Section 5 describes how CAPPLES was applied to characterise the synthetic workload used in the simulation of SICOM. The simulation process is described in Section 6. Section 7 presents some experimental results. Finally, conclusions are presented in Section 8.

2 Terminology and Migration Assumptions

In this section, we define some terms and present the migration assumptions that we consider in this work.

2.1 Terminology

Services. An application can be modelled as a hierarchy of tasks, each one with a specific goal [17]. Tasks can be decomposed into subtasks, that can be decomposed in subsubtasks, and so on. Thus, tasks can be specified at different levels of abstraction. *Services* are top-level tasks composed of a set of conventional actions performed by their subtasks over the system. As a task, a service also has a specific goal.
On-line transactions. There are subtasks that represent each interaction of the user with the application through the use of interactive devices, such as displays, keyboards, and mice. On-line transactions are less abstract tasks than services. In fact, an *on-line service* can be decomposed into many on-line transactions.
Batch jobs. Services not associated with interactive devices are called batch jobs. Users do not interact directly with services of this category. In fact, printed reports are the usual output of batch jobs. Further, batch jobs usually process large amounts of data. Batch jobs can also be called *batch services.*

2.2 Migration Assumptions

Two basic assumptions must be considered when migrating a legacy system:

1. *The migration of a legacy system is time-consuming.* The migration of legacy systems requires planning and preparation. In fact, the resources of the new target system environment must be installed and tested. Moreover, technical people must be trained to operate these resources, and users must be trained to use the target system. Worst, users usually can only be trained after the technical people are ready to operate the new environment.
2. *Developers know the target system.* The target system developers know the individual behaviour of the services provided by the target system. Indeed, it is expected that these services were individually tested with success. Thus, the developers know what services in the target system are the most likely to present performance problems.

3 The CAPPLES Method

3.1 The Scope

CAPPLES, as presented here, is a method general enough to deal with many different migration scenarios. However, the method has a well-defined scope where it can be applied, as described below.

Workloads are generated due to services of three categories: (1) interactive requests, (2) on-line transactions, and (3) batch jobs [19]. Considering this workload composition, CAPPLES is a method for evaluating the performance of target systems running services of any category, *but where the on-line services are suspected to have performance problems.* In fact, batch jobs rely basically on the servers' resources (e.g., CPU, I/O subsystem, operating system). On-line transactions, however, rely on the whole environment, i.e., they rely on the servers' resources, like batch services, but also on client computers' resources, local networks, remote links, network devices (e.g., routers, hubs, switches), and local and distributed softwares (e.g., transaction monitors, operating systems, network protocols). Therefore, batch job performance problems tend be solved easier than on-line transaction performance ones. Further, interactive requests are not used intensively in legacy systems. Indeed, they are usually provided by stand-alone applications to perform specialised services. Therefore, interactive request performance problems can be solved in an ad-hoc manner.

According to Jain [16], there are three approaches to predicting a system's performance: (1) analytical methods, (2) experimental studies, and (3) simulations. Considering the problem of migrating legacy systems, the use of analytical methods requires many simplifications and assumptions about the target system. Thus, the resulting model would possibly not model the target system with accuracy. Experimental studies are also usually not appropriate. Although the target system might be fully developed, its operational environment may not be set up. Besides this, in order to fit within the precision needed in this kind of study, the experiments would have to be carried out several times, requiring the allocation of many people in order to generate the predicted workload. Worst, it is hard to guarantee repeatability in such a kind of experiment. Therefore, simulations tend to be the most appropriate approach for such studies. Indeed, simulation models are very flexible for implementing the details required. Moreover, they are flexible enough to be modified, so that many scenarios can be evaluated from a validated model.

3.2 Overview of the Method

CAPPLES is composed of 9 steps which are described bellow. Additional details are presented in Sections 4, 5 and 6, and also in [7].

Step 1: Specification of the measuring time window. Many systems have an on-line window, where the on-line transactions have higher priority than services of other categories [19]. Inspecting the scheduled routines of a legacy system and its environment (e.g., operating system tuning and transaction monitors tuning), it is possible to identify the on-line window, which corresponds to the measuring time window to be used in Step 2. The measuring time window usually covers the week-days from 9 am to 5 pm, when users are effectively interacting with the legacy system. However, its specification depends essentially on the organisation's activities. For example, the measuring time window for supermarkets can have extended working hours, including weekend-days.

Step 2: Measurement of the legacy system and specification of the simulation time window. In order to be able to collect the data required to evaluate the performance of the legacy system, system monitors must be chosen to be used along the measuring time window. Although it is possible to collect hundreds of different types of performance data, CAPPLES only requires the following: the mean response time (MRT) of the legacy system on-line transactions to be used for comparison with the simulated MRT of these same transactions (Step 9), and the frequency of each service during the measuring time window, summarised by hour (or half-hour, quarter-hour, etc.). These data help the identification of the peak hour (or half-hour, etc) of the legacy system in terms of its utilisation. This peak hour is the *simulation time window* that will be used in Steps 7 and 9. There are two aspects that should be observed: (1) the legacy system can be monitored for as long as possible, but the workload profile tends to be almost the same, week by week (this happens since only high-frequent services are counted, as described in Step 3); (2) the duration of the simulation time window should be large enough to accommodate the occurrence of services that are relevant for the composition of the synthetic workload as described in Step 5.

Step 3: Identification of the relevant services in the legacy system. CAPPLES adopts the assumption that usually 20% of the programs are responsible for 80% of the system workload [14]. The proportion could not be exactly 20/80, but the idea is to identify a reduced set of services that are responsible for a significant part of the system workload. Once again we have a trade-off between the required precision of the performance prediction and the available time for performing the study. It is important, however, to observe that a general understanding of the services selected is required. However this is a very difficult task since legacy services are usually not well-documented. Based on this trade-off, services that compose the synthetic workload are selected, as described in Step 5. It is important to respect the selection order. Services cannot be selected if they are less frequent than other services not selected.

Step 4: Mapping of the relevant services from the legacy system to the target one. In CAPPLES, the intensity of the synthetic workload is provided by the legacy system and the resource demand[1] of each service is provided by the target system. Therefore, it is required to map the on-line transactions from the legacy system to the target one. However, a direct on-line transaction mapping is usually not possible in practice, since it is very difficult to recognise the correct equivalence between the transactions in both systems. Thus, in our method, this mapping is carried out indirectly through on-line services, as shown in Fig. 1. This is possible because on-line services are higher-level abstractions (see Section 2.1) that encapsulate conventional actions that must exist in both systems.

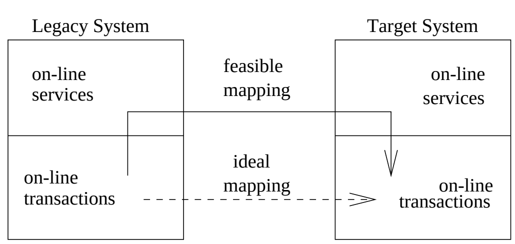


Fig. 1. Mapping of on-line transactions.

[1] The amount of time the resource is allocated exclusively to the service [19].

We notice, however, that mapping on-line transactions into on-line services in the legacy system is not usually a trivial task. The reason for this difficulty is that legacy systems often have out-of-date documentation and few people have a comprehensive knowledge of how they have been implemented. On other hand, users know very well how to operate legacy systems and developers have a complete understanding of the target systems. Therefore, the collaboration between users and developers usually solves the mapping between on-line services. Additionally, the mapping of on-line services into on-line transactions in the target system can be solved by the development team.

Step 5: Generation of a synthetic workload for the target system. At this step the synthetic workload is partially ready, since the intensity of the services was identified in Step 2 and the composition of the workload was identified in Step 3. The other service demands such as CPU time, local and remote network utilisation, and hardware specification must be taken from the target system and its environment. The exact composition of the service demands depends on the system's environment considered (e.g., database management system, local network, remote network, etc.), the composition of the workload (e.g., on-line services, batch jobs, spooler jobs, etc), and also the performance monitors available in the legacy and target system environments. In fact, using the identified composition and intensity of the workload, the remaining workload characterisation (and even the remaining capacity planning and performance analysis method) is the same as for the traditional methods.

Step 6: Modelling the target system. Modelling the target system requires a precise comprehension of it. Hopefully, the target system developers can help performance analysts in this task. Therefore, using a general purpose discrete event simulator (e.g., SES/*workbench* [22], SMPL [18]), the target system can be modelled by describing its characteristics that affect the behaviour of the on-line transactions. Three important aspects should be observed at this step: (1) High-level detailing is not usually required when modelling the target system. Indeed, the best approach is to start with a simple model, increasing the details gradually, until the model is considered validated (this is the calibration process in Step 7); (2) The model must contemplate all services identified during the specification of the simulation time window (Step 3). Therefore, if there were batch jobs, or even interaction requests, running during the simulation time window, they must be considered in the simulation; (3) Every resource that can be temporarily blocked by a service (e.g., database pages [1], LAN's, printers) or that have high probability of being saturated (e.g., remote links with small bandwidth) is a candidate to be modelled.

Step 7: Calibration and validation of the target system model. As CAPPLES focuses on on-line transactions, the model validation is based on the MRT of the on-line transactions. Therefore, a simulation model is considered validated if the MRT of the modelled on-line transactions corresponds to the measured MRT of the same transactions in the target system. It is important to observe that the workload must be a relevant one, and that the measured environment must be as complex as the real target system environment. This environment, however, does not need to have the same size as the real environment. Indeed, a small synthetic workload, in terms of active users, can be larger than real workloads, since the synthetic workload intensity can be much larger than the real workload intensity. To achieve this result, we just need to set the virtual user think-time as close as possible to the response time.

Step 8: Workload prediction. The workload generated in Step 5 corresponds to the current demand of the organisation. In this step, this workload should be conditioned in order to reflect the moment that the target system will be in charge of all the services. Workload forecast techniques that take into account business units of the organisation can be used [19].

Step 9: Target system performance prediction. Having designed the predicted workload (Step 8), it can then be submitted to the validated simulation model (Step 7), generating all required information concerning the behaviour of the target system. The advantage of using a simulation model is that it can be easily modified, so that different scenarios can be evaluated.

In the next section we discuss the use of CAPPLES to predict the behaviour of a real target system.

4 SICOM: A Case Study for CAPPLES

CAPPLES was used to predict the behaviour of a target system, called SICOM, which was developed to replace the commercial system operating at COPASA-MG since 1985. This commercial system, called HP07, is a mainframe-based legacy system composed of about 3,500 programs written in COBOL and Natural. As a legacy system, the HP07 system was not fulfilling the organisation needs. Thus, a migration effort was initiated to develop a completely new system, the SICOM.

After five years of development, SICOM was ready to replace the HP07. However, as this system is a mission-critical one for COPASA-MG, its replacement should be carefully carried out. In fact, the HP07 system is a mission-critical system since it is in charge of all commercial tasks and some of the operational tasks of an organisation responsible for water supply for more than 5 million customers. As the organisation is fully adapted to work according to the response times of the legacy system, CAPPLES was used to predict the response times of SICOM's on-line transactions. Therefore, the expected response times of the target system were known before the migration took place, so that the organisation could, if needed, adjust its services to work according to the predicted response times.

SICOM has 1,291,087 lines of code, divided in about 6,500 programs, and was entirely developed in Natural using the ADABAS database management system [24]. It was developed to operate in a distributed environment. In contrast with HP07, which processed data from the whole state of Minas Gerais in a centralised mainframe, SICOM will operate in a distributed environment in seven distinct regions of Minas Gerais. In this study, we have analysed the behaviour of SICOM in the Northern Region's environment. However, the results of this work can be extended to the other regions in a simple manner. The Northern Region is divided in three districts named Montes Claros, Janaúba, and Januária. SICOM will operate in Montes Claros, and users from Janaúba and Januária will use the system through `telnet` sessions. The three districts are interconnected using X.25 links. Fig. 2 illustrates this operational environment.

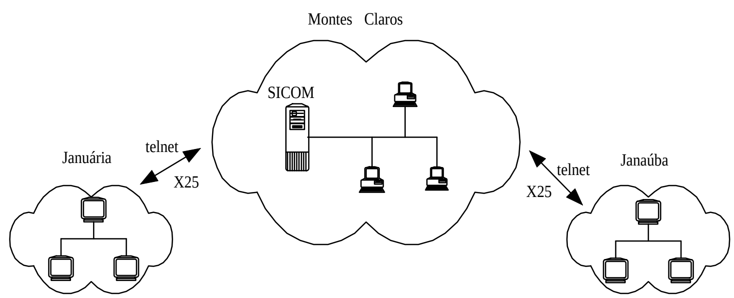


Fig. 2. The Northern Region environment.

5 Workload Characterisation

In this section, we describe how CAPPLES was used to predict the workload that will be submitted to SICOM. Each following subsection is related to a specific step of CAPPLES.

5.1 The Measuring Time Window

The first issue to address in CAPPLES is the specification of a measuring time window in which the system will be analysed. The measuring time window specified for SICOM corresponds to the commercial hours of COPASA-MG, which includes all weekdays from 8 am to 6 pm, plus two extra hours, 7 am to 8 am, and 6 pm to 7 pm. These extra hours are due to the number of on-line services executed during such hours, as reported by the IBM CICS/MVS [15], which is the on-line transaction monitor in use.

5.2 Measures Taken from the Legacy System

Once the measuring time window is defined, the legacy system must be monitored. In CAPPLES, two measures are of vital importance: (1) the frequency of execution of the legacy system programs and (2) the average response time of these programs.

Another information considered in this case study, mainly due to the distributed architecture of SICOM, was the identification of the terminal of each executed program. This information allowed us to find out the data belonging to the Northern Region and, therefore, use them to predict the workload for that region. As we suppose that the legacy system is still operational, these measures can be easily taken using any on-line transaction monitor available. For instance, in our case study we have used the CICS Manager monitor [5] to collect these performance measures.

5.3 Identification of the Relevant Services in the Legacy System

Having identified the most frequently executed programs, it is necessary to find out which services these programs are related to. Usually, this can be accomplished through interviews with users of the legacy system. Alternatively, the source code of the legacy system can be analysed.

Suppose, for instance, that the programs PT78 and PT88 were the most frequently executed programs in HP07. Analysing the source code reveals that these programs refer to, for example, the "order entry" service. So, the frequency of execution of the programs PT78 and PT88, in this case, is about the same as the frequency in which an order entry is executed in the system. In this way, we can find out the frequency of execution of the most relevant services in the legacy system.

An important issue to address in this step is the number of services that must be analysed and modelled with CAPPLES. This will vary from system to system. In this case study, we have used 22 services, corresponding to about 30% of the total processing activity of all systems run on the COPASA-MG's mainframe. In terms of the HP07 system, these 22 programs correspond to about 70% of the use of this system. Adding more services would not be helpful, because these new services would contribute little to the overall statistic. We have observed that the inclusion of the next most executed service (i.e., the 23rd) would correspond to an increase of only 0.6% the representation of the overall mainframe usage.

5.4 Mapping of Relevant Services

Once the most relevant services and their execution frequency are determined, it is necessary to map these services from the legacy system to the target one. As these services are high level abstractions of the real system's transactions (e.g., an order entry in the system), this mapping can be easily accomplished through interviews with the developers of the target system.

At this point, after the complete mapping of the relevant services, we have the frequency of execution of these services in the target system. However, as the target system might have added new functions, it is difficult to find out how these services will be executed in the new system. Possibly there are different ways of performing the same service in the target system (we call them *use cases*). Therefore, it is necessary to analyse every possible use case in the target system. Further, we need to distribute the frequency of execution of the services between these use cases. Sometimes the data measured in the legacy system can help at this step.

In the case study, we jumped from 22 services to 43 use cases in the target system. Fig. 3 illustrates the process of identifying and mapping services from the legacy to the target system.

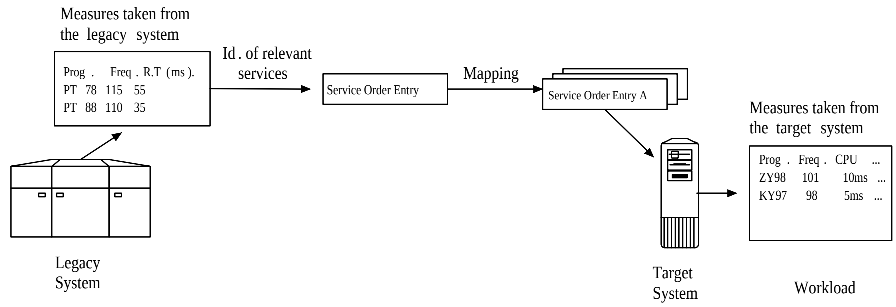


Fig. 3. The workload characterisation process.

5.5 Generating the Synthetic Workload

In order to generate the workload that will be submitted to the target system, two characteristics must be determined: its intensity and its services' demands. Up to this point, the intensity of the workload has already been defined by the prior steps of CAPPLES.

The measurement of the services' demands of the relevant services can be accomplish in a straightforward manner, since we assume that the target system is fully developed. It is simply a matter of using accounting tools of the operating system (e.g., acct, entstat) for measuring the services' demands while the target system is running.

6 Simulation Model

A general purpose simulation tool, SES/*Workbench* [22], was used to build the simulation model. The following components of the target system were modelled (see Fig. 4):

Workload. This corresponds to the predicted workload intensity and services' demands generated with CAPPLES.

Natural Subsystem. Each program present in the workload model competes for some system's resources (e.g., CPU, LANs). The Natural Subsystem represents this use of resources by the applications.

ADABAS Subsystem. The ADABAS DBMS was modelled since it uses CPU cycles and makes accesses to the I/O subsystem.

LANs and WANs. Ethernet and X.25 models were also created since they can be sources of contention and increase the response time of the system.

CPU and I/O Subsystem. The system's CPU and I/O subsystem were the main resources modelled. We assumed that enough main memory was available, so it would not influence the system's performance.

Initially, the workload component generates transactions[2] following the predicted workload. These transactions, as they enter the Natural Subsystem model,

[2] The term transaction, here, refers to simulation transactions that can be considered as threads in the simulator.

consume CPU cycles, and use the ADABAS subsystem model to make the I/O subsystem accesses. The ADABAS subsystem also uses some CPU cycles. Finally, the transactions travel through the LAN and WAN models and terminate. The cost and quantity of these CPU and I/O subsystem accesses are defined by measures taken from the target system, as considered by Step 5 of CAPPLES. Fig. 4 gives a high level illustration of the model developed and includes only its main components. However, batch jobs and `ftp` services[3] were also modelled, since they were identified in the simulation time window. An in-depth description of the whole simulation model developed can be found in [7].

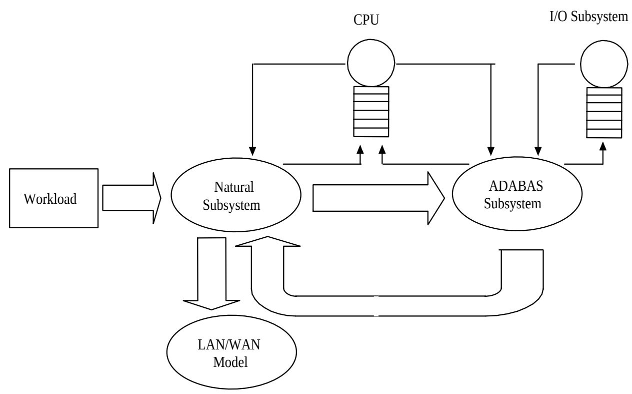


Fig. 4. The top-level diagram of the SICOM simulation model.

In order to validate the simulation model built and make sure it represented the real system accurately, some experimental results were compared to the simulation output.

The experiment consisted in measuring the response time of SICOM when exposed to different synthetic workloads. Three sets of measures were taken, representing workloads generated by 8, 14 and 24 simultaneous clients. In order to measure the response times of the on-line transactions, three possible approaches were considered:

Modification of the SICOM source code. The target system source code could be modified to register the system response times. The drawback in this case is that the network time is not measured.

Insertion of a net spy. Another choice is to insert a network spy that receives the client requests and forwards it to the server and vice-versa saving the response times in the process. In this approach, new network packages are created, generating considerable system overhead.

Using a modified telnet client. In this solution, a modified `telnet` client generates the workload and saves the send request time. When the answer

[3] This new category was created since ftp services are a mixture of on-line and batch services, in terms of performance demands, as found in SICOM.

is received, the response time is computed and logged in the memory of the client computers. This was considered the best approach, since it mimics the real world actions in a repeatable manner.

Using the third approach, we compared the experimental results with the simulation output. After four significant modifications in the simulation model (and others not so significant), the difference between the measured and simulated results were no higher than 5%.

The experiment carried out for the simulation validation was one of the most time demanding tasks in the whole study, mainly due to difficulties in generating the workload and logging the response times. This shows that the use of simulation instead of experimental methods is really a good choice.

7 Experimental Results

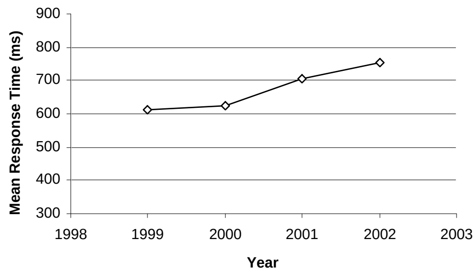
In this section, we describe the results of two experiments carried out using the validated target system model. The results of the first experiment show that the target system can cope with its expected workload. The second experiment produced a target system profile in terms of resource utilisation. In both experiments, the confidence interval was built upon the on-line transaction's average response time. All of our results are expressed as a mean value plus or minus a few percentage points, since we built confidence intervals of at least 90%.

The results of the first experiment show that the target system will work, provided that its response time is smaller than the response time of the legacy system. Indeed, organisations are adapted to deal with the response time of the legacy systems (e.g., number of concurrent users, number of required transactions per hour, etc.) and a significant increase of the response time could require changes in the organisation that they cannot cope with.

The graphic of Fig. 5 (a) shows the simulated mean response time of the target system predicted by year. The graphic also shows that the simulated mean response time of the target system will be 612.07 ms in 1999, which is smaller than the 960.00 ms of the legacy system. This means that, in general, the target system is a real improvement in terms of performance. The mean response time, however, is not enough to know if we can shutdown the legacy system, since essential transactions could have unacceptable response times, e.g., 10 or more minutes. As we work with a simulation model, we can trace the response time of all transactions, producing an histogram, as shown in Fig. 5 (b).

Having solved the main problem, the simulation model can provide additional information about the behaviour of the target system, such as the resource utilisation shown in Table 1. There, we notice that the resources allocated to the target system are far from being saturated. However, some attention should be spent to the database server's CPU, since this is the resource with the highest level of utilisation. Other resources, such as local networks and the database server's I/O subsystem, are extremely lightly loaded.

This resource utilisation table is only a snapshot of the target system environment, at migration time. As the utilisation of computer resources is not



(a)

Response Time Range (ms)		Percentage
from	to	(%)
0.000	100.000	8.10
100.000	300.000	31.55
300.000	500.000	18.78
500.000	900.000	18.05
900.000	1300.000	11.47
1300.000	2000.000	8.46
2000.000	3000.000	2.94
3000.000	+infinity	0.64

(b)

Fig. 5. Mean response time (a) and histogram (b) of SICOM's transactions.

Resource	Utilisation (%)
Database server's CPU	48.184
Database server's I/O subsystem	1.557
Montes Claros local network	0.071
Januária's local network	0.011
Janaúba's local network	0.007
Remote link between Montes Claros and Januária	20.347
Remote link between Montes Claros and Janaúba	18.882

Table 1. Resources utilisation.

linear, the simulation model must be used to predict the behaviour of the target system after the migration.

A third experiment was carried out to compare the performance of the target system with its hypothetical client-server version, showing the benefits of the migration from the multi-user system to a client-server environment. Details of the modified version of the target system model and the results achieved are available in [7].

8 Conclusions

This paper presented CAPPLES, a method for evaluating the performance of target systems during the migration of legacy systems. The method has been shown to be effective in practice since it was successfully used to evaluate a target system with more than one million lines of code, as described in the case study presented here.

The performance analysis of the case study spent 8 months, which is an acceptable time since the development of the target system took 5 years and the planning for its migration 10 months. In fact, the performance analysis took place in parallel with the migration planning. The evaluation of the case study took 8 months, but part of this time was invested in enhancing CAPPLES. Thus, if we were able to apply CAPPLES from the very beginning, we believe that the evaluation time could have been reduced to around 6 months. Moreover, as a significant part of this time was spent designing, calibrating, and validating the target system model, having the support of a simulation specialist would reduce this time even further.

As we can see from our case study, using a method like CAPPLES is key for the success of the migration process, since performance problems can be identified and solved before the target system becomes entirely operational. Moreover, the target system environment can be better specified, and computational resources requirements more accurately assessed.

Finally, carrying out a performance analysis study during the migration of a legacy system is important not only for managers, but also for developers and users. Indeed, after the first year of development of a new system, the idea that something is going wrong can threaten the project. Spending 5 years (or even more) developing the target system, even the developers loose their confidence in the performance of the new system. Therefore, such a study is important to motivate everyone involved in the migration of a legacy system, as well to provide the organisation with accurate feedback on the inherent complex problems concerning the migration of legacy systems. In the case of SICOM, this study showed that the system's environment had been well dimensioned and that, apart from the database server's CPU, the resources allocated to the target system are far from being saturated.

Acknowledgements. The authors would like to thank Berthier Ribeiro-Neto and Virgílio Almeida for many valuable comments during this work, Daniela Seabra for her assistance during the experiments, and Norman Paton for his comments on a first draft of this paper. The first author is sponsored by Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq (Brazil), grant 200153/98-6.

References

1. R. Agrawal, M. Carey, and M. Livny. Concurrency control performance modeling: Alternatives and implications. *ACM Transactions on Database Systems*, 12(4):609–654, December 1987.
2. P. Aiken, A. Muntz, and R. Richards. DoD legacy systems: Reverse engineering data requirements. *Communications of the ACM*, 37(5):26–41, 1994.
3. K. Bennett. Legacy systems: Coping with success. *IEEE Software*, 37(5):26–41, 1995.
4. J. Bergey, L. Northrop, and D. Smith. Enterprise framework for the disciplined evolution of legacy systems. Technical Report CMU/SEI-97-TR-007, Carnegie Mellon University, Pittsburgh, PA, October 1997.
5. Boole & Babbage, San Jose, CA. *CICS Manager Performance Manager User Guide, Version 3, Release 4*, June 1995.
6. M. Brodie and M. Stonebraker. *Migrating Legacy Systems: Gateways, Interfaces, and the Incremental Approach.* Morgan Kaufmann, San Francisco, CA, 1995.
7. P. Pinheiro da Silva. A study on the migration of centralised legacy systems to distributed enviroments. Master's thesis, Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil, June 1998. (In Portuguese).

8. G. Dedene and J. De Vreese. Realities of off-shore reengineering. *IEEE Software*, 12(1):34–45, January 1995.
9. E. Merlo et al. Reengineering user-interfaces. *IEEE Software*, 12(1):64–73, January 1995.
10. L. Markosian et al. Using an enabling technology to reengineer legacy systems. *Communications of the ACM*, 37(5):58–70, 1994.
11. D. Ferrari, G. Serazzi, and A. Zeigner. *Measurement and Tunning of Computer Systems*. Prentice Hall, Englewood Cliffs, NJ, 1983.
12. S. Goldberg, A. Pegalis, and S. Davis. *Y2K Risk Management; Contingency Planning, Business Continuity, and Avoiding Litigation*. John Wiley & Sons, New York, NY, 1999.
13. J. Gunther. Capacity planning for new applications throughout a typical development life cycle. In *Proceedings of the Computer Measurement Group Conference*, Nashville, TN, December 1991.
14. J. Hennessy and D. Patterson. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann, San Francisco, CA, 1996.
15. IBM Corporation, Mechanicsburg, PA. *CICS/MVS Version 2.1.2 Facilities and Planning Guide*, March 1991.
16. R. Jain. *The Art of Computer System Performance: Analysis, Techniques for Experimental Design, Measurement, Simulation and Modeling*. Wiley, New York, NY, 1991.
17. B. Kirwan and L. Ainsworth. *A Guide to Task Analysis*. Taylor & Francis, London, UK, 1992.
18. M. MacDougall. *Simulating Computer Systems: Techniques and Tools*. MIT Press, Cambridge, MA, 1987.
19. D. Menascé, V. Almeida, and L. Dowdy. *Capacity Planning and Performance Modeling: From Mainframes to Cliente-Servers Systems*. Prentice-Hall, Englewood Cliffs, NJ, 1994.
20. J. Ning, A. Engberts, and W. Kozaczynski. Automated support for legacy code understanding. *Communications of ACM*, 37(5):50–57, 1994.
21. G. Pernul and H. Hasenauer. Combining reverse with forward database engineering - a step forward to solve the legacy system dilemma. In *Proc. of 6th DEXA Conference*, pages 177–186, London, UK, 1995.
22. Scientific and Engineering Software, Austin, TX. *SES/*workbench*, Technical Reference, Release 3.1*, 1996.
23. H. Sneed. Planning the reengineering of legacy systems. *IEEE Software*, 12(1):24–32, January 1995.
24. Software AG, Darmstadt, Germany. *ADABAS Concepts and Facilities Manual, Version 6.2*, August 1997.
25. I. Sommerville, J. Ransom, and I. Warren. A method for assessing legacy systems for evolution. Technical Report TR/07/97, Computing Department, Lancaster University, Lancaster, UK, 1997.
26. C. Wilson. Performance engineering-better bred than dead. In *Proceedings of the Computer Measurement Group Conference*, pages 464–470, Nashville, TN, December 1991.