A Cost-Oriented Approach for Infrastructural Design

Danilo Ardagna
Politecnico di Milano
Dipartimento di Elettronica e Informazione
Via Ponzio 34/5, 20133 Milano, Italy
+39-02-2399-3561
ardagna@elet.polimi.it

Chiara Francalanci
Politecnico di Milano
Dipartimento di Elettronica e Informazione
Via Ponzio 34/5, 20133 Milano, Italy
+39-02-2399-3457
francala@elet.polimi.it

Marco Trubian
Università degli Studi di Milano
Dipartimento di Scienze dell'Informazione
Via Comelico 39, 20135 Milano, Italy
+39-02-503-16230
trubian@dsi.unimi.it

ABSTRACT

The selection of a cost-minimizing combination of hardware and network components that satisfy organizational requirements is a complex design problem with multiple degrees of freedom. Decisions must be made on how to distribute the overall computing load onto multiple computers, where to locate computers and how to take advantage of legacy components. The corresponding optimization problem not only embeds the structure of NP-hard problems, but also represents a challenge with a well-structured heuristic approach. A scientific approach has been rarely applied to cost minimization and a rigorous methodological support to cost issues of infrastructural design is still lacking. The methodological contribution of this paper is the representation of complex infrastructural design issues as a single cost-minimization problem. The problem is decomposed in four intertwined cost-minimization sub-problems; optimization is accomplished by sequentially solving these sub-problems with a heuristic approach and tuning their solution with a final tabu-search step. Results indicate that decomposition significantly reduces optimization time and solutions are also closer to the global optimum if results are compared to those identified without prior decomposition. Cost reductions are also significant when practicioners' solutions, obtained by applying simplified design rules from the professional literature, are considered.

Categories and Subject Descriptors

C.2.4 [**Distributed Systems**]: Client-server, Distributed applications.

General Terms

Algorithms, Performance, Design.

Keywords

Cost minimization, tabu-search.

SAC '04,, March 14-17, 2004, Nicosia, Cyprus.

1. INTRODUCTION

The information technology (IT) infrastructure is comprised of the hardware and network components of a computer system [29, 19]. Since hardware and network components cooperatively interact with each other, the design of the IT infrastructure is a systemic problem. The systemic objective of infrastructural design is the minimization of the costs required to satisfy the computing and communication requirements of a given group of users [14]. In most cases, multiple combinations of infrastructural components can satisfy requirements and, accordingly, overall performance requirements can be differently translated into processing and communication capabilities of individual components. These degrees of freedom generate two infrastructural design steps: the selection of a combination of hardware and network components and their individual sizing (see Figure 1).

Cost-performance analyses are executed at both steps. Performance analyses receive a pre-defined combination of components as input and initially focus on the application of mathematical models to define the configuration of each component [19]. Performance bottlenecks are then identified at a system level and removed by re-sizing specific components that constrain system-level performance. Conversely cost analyses start at a system level, to identify a combination of components that minimizes overall costs, which is initially calculated from rough estimates of individual components' configurations and corresponding costs. The evaluation of costs of individual components is subsequently refined based on more precise sizing information from performance analyses (see Figure 1). Due to this interdependence between cost and performance analyses at both design steps, the overall infrastructural design process is iterative.

The goal of this paper is to support the cost-oriented design of modern IT infrastructures with a rigorous optimization approach to help the scientific verification of empirical design rules. Infrastructural design alternatives are organized within a methodological framework and are provided a formal representation suitable for optimization. Four intertwined cost-minimization sub-problems are identified: two set-partitionings, a set-packing and a min k-cut with a non linear objective function. Optimization is accomplished by sequentially solving all sub-

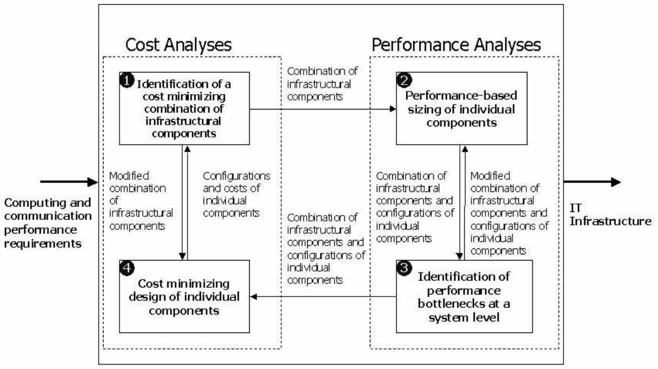

Figure 1 – The infrastructural design process.

problems with a heuristic approach and finally tuning their solution with a final tabu-search step. The aim is to evaluate a large number of alternative solutions and find a candidate minimum cost infrastructure that can be in the following analyzed applying fine-tuning performance evaluation techniques.

2. RESEARCH BACKGROUND AND MOTIVATION

The historical cost-minimizing design principle was *centralization*, which was advocated to take advantage of hardware scale economies according to Grosch's law [11]. A further attempt to formulate a general design principle has been made in the mid '80s, when Grosch's law has been revised as "It is most cost effective to accomplish any task on the least powerful type of computer capable of performing it" [8]. *Decentralization* and its operating rule, referred to as *downsizing*, became the methodological imperative for cost-oriented infrastructural design.

Although academic studies have challenged the generality of the decentralization principle [10, 14], the empirical recognition of the cost disadvantages of decentralization has only occurred in the '90s with the observation of client-server costs. Addressing this failure, empirical studies have showed that initial acquisition expenses represent at most 20% of the total cost of a computer over its life cycle [30] and as a consequence, the minimization of acquisition costs does not deliver the most convenient infrastructural design solutions. The concept of "total cost of ownership" (TCO) has been introduced and defined as the summation of both investments and management costs of infrastructural components [9]. It has been observed that while decentralization reduces investment costs, it increases management costs, due to a more cumbersome administration of a greater number of infrastructural components [21, 27]. *Recentralization* has been thereafter considered to reduce management costs. The rationale for recentralization is that the client-server paradigm can be extended to allow multiple machines to share computing load [7]. Applications can be designed to be split into multiple modules, called *tiers*, each of which can be allocated on a different machine [19]. Multi-tier applications give rise to multi-tier infrastructures, that offer greater flexibility to implement the most convenient load sharing among multiple machines. *Thin clients* (TCs) are currently proposed as a less expensive alternative to personal computers that can be exploited through a recentralization initiative (see Table 1, [23]). Thin clients have lower computing capacity than PCs, which is sufficient for the execution or the emulation of the presentation tier, but requires the recentralization of the application logic on a server and have management costs 20-35% lower than personal computers [22, 28]. Furthermore, the *Independent Computing Architecture* (ICA) and *Remote Desktop Protocol* (RDP) standards allow remote access to the application logic by traditional PCs. This translates into hybrid configurations of PCs which execute only a subset of client and monolithic applications which will be referred to in the following as *hybrid fat clients* (HFCs). An application tier can also be simultaneously allocated on multiple coordinated machines, known as *server farm* [19]. Each computer within a server farm autonomously responds to a subset of service requests addressed to the application tier, thus sharing the overall computing load with other computers within the same farm. This load sharing allows downsizing and reduces acquisition costs. Furthermore, it has limited negative effects on management costs, since server farms are equipped with software tools that allow the remote management and simultaneous administration of all servers [20].

Another important concern in infrastructural design is the reuse of existing components, referred to in the following as *legacy systems*. Legacy systems have often a high residual economic value and, thus, their reuse becomes a relevant choice in infrastructural design and can shift cost-trade-offs [4]. Furthermore, they can be upgraded and their life cycle can be extended over a significantly longer period of time with limited additional investments. Even if current professional guidelines generally recommend recentralization, the reuse of legacies may induce different design choices, reinforcing the need for a rigorous optimization approach.

Table 1 summarizes the infrastructural design alternatives that generate centralization-decentralization cost trade-offs. Overall, current design rules generally encourage solutions to these design alternatives that translate into a recentralization of hardware components. However, most research efforts addressing centralization-decentralization issues lack scientific rigor and only a few academic studies have attempted a more systematic analysis of cost issues in infrastructural design [10, 14].

3. DESIGN METHODOLOGY AND OPTIMIZATION ALGORITHM

From a methodological standpoint, revisiting centralization-decentralization trade-offs requires the representation of design alternatives in Table 1 as a single cost-minimization problem.

In the current work, the last design alternative in Table 1 is pre-constrained to a specific topology and standard for both local and geographical networks. LANs that connect different buildings within the same site are constrained to the extended-star topology. Different sites are constrained to be connected through an IP-based Virtual Private Network (VPN). In this way, network design is not explicitly addressed; however, the methodology includes both a sizing and a costing step for network components. This provides a necessary input for the evaluation of total infrastructural costs and allows preliminary analyses of the impact of network costs on hardware design choices. The goal of the methodology is to select a combination of infrastructural components that minimizes the TCO of the overall infrastructure while satisfying requirements. This involves an initial specification of *technology requirements*, which will be described in the next Section and a *cost-minimization process*, which is then presented in Section 3.2.

Table 1 – Infrastructural design alternatives

Macro-alternative	Sub-alternative	Description
How to distribute the overall computing load of a system onto multiple machines.	Client typology, thin vs. fat vs. hybrid fat client (HFC)	Thin clients manage the user interface of applications stored and executed remotely, while fat clients store and execute applications locally. Hybrid fat clients (HFCs) behave both as fat and thin clients depending on the specific application.
	Number of tiers	The client-server paradigm organizes applications in multiple tiers. Each application tier can be allocated on a separate machine and responds to service requests from lower tiers.
	Total number of servers	The required computing capacity can be allocated on one or multiple servers, organized as server farms, whose total number represents an architectural alternative.
	Allocation of applications	Different applications (or application tiers) can be allocated on separate computers and, vice versa, multiple applications can be allocated on the same computer.
	Reuse of legacy systems	Requirements can be satisfied by means of either new or legacy systems. Legacy computers can also be upgraded to satisfy increasing capacity requirements.
Where to locate machines that need to exchange information.	Location of servers	Servers can be located in different sites, although all servers within the same server farm must be located in the same site.
	Network topology and standards	Sites can be connected with different routing policies and through different logical and physical communication standards.

3.1 Technology requirements

Technology requirements are expressed by means of the following fundamental variables:

- *Organization sites S_i*, defined as sets of organizational resources (users, premises and technologies) located within a 1 Km distance from each other (and connected through a LAN).
- *User classes C_i*, defined as a group of n_i users using the same subset of applications, with common capacity requirements. User classes are located in an organization site, are characterized by a *think-time* (*high* or *low*) and can be associated with multiple client typology (*thin, fat* or *hybrid*).
- *Applications A_i*, defined as a set of functionalities that can be accessed by activating a single computing process. Applications are classified as *client*, *monolithic* or *server* and are characterized by computing and memory (primary and secondary) requirements.
- *Requests R_i*, defined as interactions among applications aimed at exchanging services according to the client-server paradigm. Requests are characterized by their frequency, the set of supporting server applications, corresponding CPU and disk demanding times and data exchanged for each request. Demanding times are supposed to be evaluated on a tuning system, this allows the estimate of demanding times on a different system by means of benchmarking data [18].
- *Databases D_i*, defined as separate sets of data that can be independently stored, accessed and managed. Note that DBMSs are supposed to be specified as server applications and, accordingly, databases are simply described by the size of secondary memory that they require (database are stored in the physical server that support DBMS execution).

A formal specification of technology requirements can be found in [2, 3]. The specification of sites and user classes is critical to select network components during optimization. Applications, requests and databases are the main drivers of design choices related to client and server computers. Note that for server applications multiple tier allocations can be specified. Also groups of tiers serving different requests and group of user classes assigned to thin/HFC servers can be defined. If the cardinality of a group is n, 2^n-1 different allocations of tiers or user classes are evaluated (that is, the group's power set, excluding the empty set).

The computing capacity of PCs is obtained as the maximum value of MIPS required by locally executed applications. Similarly, computing capacity of *thin/HFC servers* (servers supporting thin and HFC clients) is evaluated as the maximum value of MIPS required by client and monolithic applications that are executed remotely, considering the number of concurrent users of the corresponding user class [2, 3].

Application servers are modeled by means of a queuing network including multiple CPUs and a single disk, since current secondary memory technologies based on RAID disks can be modeled as a single resource [19]. Application servers are selected to provide computing and storage capacities that guarantee an utilization of CPUs and disk lower than 60% [19, 2]. This empirical rule of thumb, which is commonly applied in practice [18, 20], has been provided a formal validation. It has been formally demonstrated that a group of aperiodic tasks will always meet their deadlines, as long the utilization of the bottleneck resource is lower than 58% [17, 1]. Note that more detailed performance analyses should follow cost analyses to refine sizing.

3.2 Cost-minimization algorithm

The optimization problem has been split into four intertwined sub-problems, which correspond to well-structured problems of the operations research literature. A final overall re-optimization step that implements a tabu-search algorithm is also introduced, in order to improve the local optimum that is found through the isolated solution of the four sub-problems. The following sub-problems have been identified:

1) *Client optimization*: user classes are assigned to minimum-cost client computers that satisfy constraints.
2) *Server optimization*: server applications are assigned to minimum-cost machines that satisfy computing requirements and constraints.
3) *Server localization*: server machines identified by solving sub-problems (1) and (2) are allocated to sites by minimizing overall network and management costs.
4) *Reuse of legacy systems*: server machines identified by solving sub-problems (1) and (2) and assigned to organization sites by solving sub-problem (3) are replaced with legacy machines to further reduce acquisition costs.

Note that physical components, either legacy or new, are selected as the lowest-cost devices satisfying requirements. A complete specification of sizing rules applied to select physical components and the formalization of optimization sub-problems are provided in [2, 3]. A brief discussion of the four sub-problems is provided in the following.

3.2.1 *Client optimization*

In this phase, decisions are made on (a) which client computer is assigned to each user class, (b) which thin/HFC servers are necessary, (c) where thin/HFC servers are located. This optimization process has been split into three steps.

In the first step each user class C_i is assigned to a set of n_i identical client computers. All machines are located in the site where C_i resides.

In the second step the solution obtained in the first step is improved by connecting sets of thin client/HFC computers to the same server. The assignment of TCs and HFCs to servers is modeled as a *Set Partitioning Problem* [24].

In the third step a local search based algorithm attempts an improvement of the solution provided by the first two steps. Previous choices are modified by assigning a user class to a different type of client computer.

3.2.2 *Server optimization*

This sub-problem is the optimum allocation of server applications to server machines. Server applications are organized in tiers according to constraints and groups specification. Each server application or application tier has to be assigned to exactly one server (or server farm). The problem is modeled as a *Set Partitioning Problem*.

3.2.3 *Server localization*

This sub-problem is the optimum allocation of servers to sites. Two cost items are affected by the allocation of servers: WAN costs (step-wise function of the input and output bandwidth required by each site) and hardware support personnel costs. This cost-minimization sub-problem can be modeled as a min k-cut problem [16]. Since this problem is strongly NP-hard, a heuristic approach based on local search is adopted. The neighborhood of each feasible solution is defined by all solutions that can be obtained by moving a server to a different site. The search is guided by a tabu-search meta-heuristic.

3.2.4 *Reuse of legacy systems*

Each site has a (possibly empty) set of legacy machines and a set of servers defined by previous optimization steps. Each server could be replaced by one or more combinations of legacy machines. Moreover each legacy machine could be upgraded to provide higher performance. This problem can be modeled as a *Set Packing Problem* [24]. Different from legacy servers, legacy clients are supposed to be assigned to the user class that owns them.

3.2.5 *Overall re-optimization*

The decomposition of the overall optimization problem into four sub-problems does not guarantee that the final solution is a global optimum. Hence, an overall re-optimization process based on a tabu-search approach has been implemented to improve the (possibly) local optimum obtained by separately solving the four sub-problems. The move that is applied is defined as follows. A user class, say C_i, or a server application, say A_i, is disconnected from the server, say server_A, to which is currently connected located in the site S_A. A new minimum-cost server (or server farm), say server_B, is selected to replace server_A. A new minimum-cost server, say server_C, is introduced to support C_i (or A_i), which is selected by comparing the cost of allocating the server in each site different from S_A. For each site, server management costs and network communication costs are evaluated. In this way, a destination site, say S_B, is identified for server_C. At last, the possibility of discarding server_C is evaluated by connecting C_i (or A_i) to a different server in S_B. The neighborhood of a solution is defined by all solutions that can be obtained by applying this move to all user classes and to all server applications sharing a server. The search is guided by a tabu-search meta-heuristic in which only the short-term memory mechanism has been implemented.

4. EMPIRICAL VERIFICATIONS

Empirical verifications have been supported by ISIDE (infrastructure systems integrated design environment), a prototype tool that implements the methodology. The tool includes a database of commercial infrastructural components and related cost data (overall about 12,000 hardware configurations form 4 different hardware vendors). Management costs of physical components have been estimated by information provider benchmarks [23, 28, 26]. Ad-hoc surveys have been necessary to obtain VPN setup and management costs and annual fees, costs of 80 VPN configurations between 64 Kbit/s and 64 Mbit/s have been obtained. Analyses focus on two real case studies, a multi-department university and an Internet banking system. The two case studies have substantially different technology requirements; in the former, user classes are numerous and use a variety of applications, making the allocation of servers to sites a critical design alternative. In the latter the system is composed of complex multi-tier applications whose allocation on servers is particularly cumbersome, application are also CPU-intensive and the design of server farms plays a predominant role. The cost and time efficiency of the implemented problem decomposition are evaluated by comparing the methodology's output with the output of the last tabu-search optimization step operating on an initial solution obtained by applying the following professional guidelines:

1. Thin clients and HFCs are adopted whenever possible to minimize the cost of clients.
2. Server farms are implemented whenever possible and designed by selecting the smallest server that can support applications, to reduce hardware acquisition costs.
3. Applications are allocated with the maximum number of tiers allowed by constraints, to reduce hardware acquisition costs.
4. Applications that can be grouped are centralized on a single server/server farm, to reduce management costs.
5. Application servers that are not constrained to a specific location are centralized in the site that minimizes VPN bandwidth requirements, to reduce both network costs and hardware management costs.

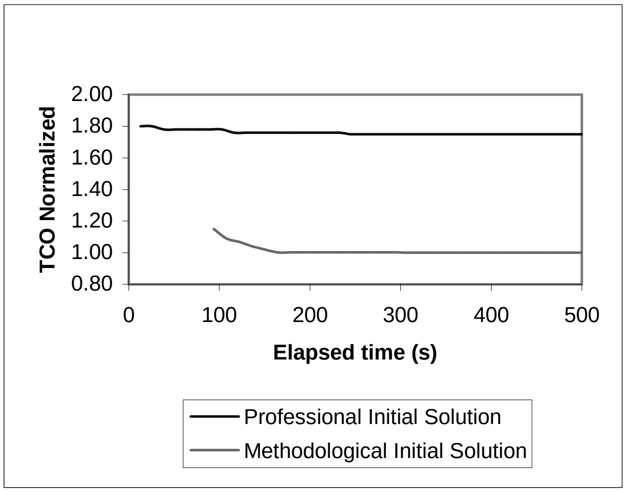

Figure 2 – TCO as a function of optimization time for the multi-department University test case.

In test cases discussed below, legacy components cannot satisfy technology requirements, as they represent available machines at implementation time and new hardware has been purchased to provide adequate capacity. Simulations have been supported by a PIII 700, W2000 workstation with 256 MB of RAM. The evaluation of the methodological initial solution is time consuming; this is shown in the following by a delay in the plot TCO vs. execution time. Despite the time required for the evaluation of the initial solution, final results obtained through the decomposition are always closer to the global optimum, as they have lower costs (20-60%). Cost reductions considerably grow (25-70%) when methodological outputs are compared with practicioners' solutions obtained applying only (1-5) design rules from the professional literature.

4.1.1 *A multi-department University*

The university is composed by 7 departments with 100 users and 3 user classes (administrative staff, software and electronic engineering researchers). User classes always require a browser, an e-mail client and an office automation suite. Software and electronic engineering researchers require an integrated development environment and a circuit simulator, respectively. The administrative staff is assigned to thin clients and researchers are assigned to HFCs. Users access local e-mail server and a web/proxy server applications; the web server also responds to requests from the Internet. E-mail and web servers can be grouped and allocated remotely. In the same way, all user classes can be allocated on the same thin server; server farm can support web servers and thin servers.

Figure 2 shows the value of TCO as a function of optimization time, TCO of intermediate solutions is normalized to the TCO of the optimal solution found. The optimal solution dismisses legacy e-mail servers, user classes are centralized on the same thin-server within each site (legacy upgraded web servers) and e-mail and web applications are associated with dedicated physical servers, that is the solution is fully distributed; reusing legacies reduces TCO by 5%.

Figure 2 shows that the decomposition makes the overall optimization step highly efficient, since the TCO of the final solution is 47% lower than the TCO of the final solution obtained without prior decomposition. The overall optimization step further improves the initial solution by 12%. If the methodological solution is compared with the infrastructure actually designed and implemented by technology experts, cost savings are about 65%.

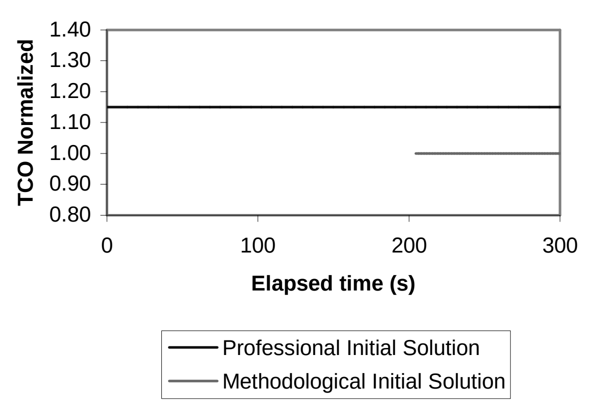

Figure 3 – TCO as a function of optimization time for the Internet banking test case.

4.1.2 *An Internet banking system*

The system is distributed in 3 data centers and supports 250,000 Internet users. Each data center hosts the following applications:

- a web server
- a servlet engine
- an application server
- a relational DBMS system which stores historical data on stock quotes
- an object-oriented database system which stores users data

Internet users are classified into three categories *active*, *moderate* and *sleepy users* depending on the average frequency of their transactions [15]. The user mix is composed by 50% active users, 30% moderate users and 20% sleepy users. The system is available to users 24 hours per day and 7 days a week. Users issue two types of requests, information retrieval and transaction execution, with a 10 to 1 ratio [15, 31, 6]. All server applications can be supported by a server farm, three different allocations of application tiers are allowed: a 5-tier allocation, which assigns each server application to a single tier and two 4-tier allocations which assign the servlet engine to the same tier of either the web server or the application server. Web applications are grouped and executed remotely. Database servers are replicated in all sites to increase fault tolerance. Originally, the Internet site was hosted in only one site and supported about ¼ of customers, previous installed components are considered for reuse as legacy.

Figure 3 shows the value of TCO as a function of optimization time. Legacy components are dismissed, database servers are replicated in all sites, according to design constraints, while web applications are centralized in one site. Web servers are centralized in one server farm; servlet engines and application servers are allocated on the same server farm, but applications serving users of different sites are allocated on separate servers. This contrasts against professional guidelines suggesting the allocation of applications with the maximum number of tiers and the centralization of applications of corresponding tiers ([7, 29, 25, 12, 13]).

Note that both the professional and the methodological solution are local optima, as the overall optimization step of the algorithm does not improve costs, irrespective of changes in tabu-list parameters. However, results show that the decomposition is still effective, as it enables a 20% reduction of TCO. Overall, the methodological solution has a TCO 27% lower than the TCO of the solution actually designed and implemented by the bank's technology experts.

5. CONCLUSIONS

Results indicate that the decomposition of the overall NP-hard problem into four sub-problems significantly reduces optimization time. Solutions are also closer to the global optimum, as they have lower costs than those identified with a local-search approach without prior decomposition (20-60%). Cost reductions considerably grow (25-70%) when methodological outputs are compared with practicioners' solutions obtained by applying simplified design rules from the professional literature.

Future work will consider the integration of the cost-oriented methodology with traditional performance analyses, which provide more precise sizing information of infrastructural components. The range of design alternatives will also be completed, by extending the methodology to include network design alternatives.

6. ACKNOWLEDGMENTS

This work has been partly supported by IBM as part of Equinox international project. Particular thanks are expressed to Gianfranco Bazzigaluppi and to Li Zhang, Zhen Liu and Cathy Xia for their feedback on the early versions of ISIDE. Particular thanks are expressed also to Andrea Molteni, Simone Amati and Alessandro Strino for their assistance in data collection and development activities.

7. REFERENCES

[1] Abdelzaher, T. F., Shin K. G., Bhatti N. Performance Guarantees for Web Server end-systems: a Control-theoretical approach. IEEE transactions on parallel and distributed systems. 13(1), 80-96, 2002.

[2] Ardagna, D. and Francalanci, C. A Cost-oriented Methodology for the Design of Web based IT Architectures. SAC2002 proceedings (17th ACM Symposium on Applied Computing). Madrid, March 2002.

[3] Ardagna, D., Francalanci, C., Trubian, M. A Multi-model Algorithm for the Cost-oriented Design of the Information Technology Infrastructure. ECIS2003 proceedings (11th European Conference on Information Systems). Naples, june 2003.

[4] Bisbal, J., Lawless, D., Bing Wu, Grimson, J. Legacy Information Systems: Issues and Directions. IEEE software, 16(5) 103 –111, 1999.

[5] Cahonn B., Mckinley k. S., Lu, Z. Evaluating the performance of distributed architectures for information retrieval using a variety of workloads, ACM Transactions on Information Systems 18(1),1-43, 2000.

[6] Cappiello, C., Francalanci, C., Pernici, B. A model of Data Currency in Multi-channel Financial Architectures. In proceedings of the seventh international conference on information quality (ICIQ '02), Boston, Ma, Usa, 2002.

[7] Dewire D. T. Second-generation Client/server Computing. McGraw Hill, 1997.

[8] Ein-Dor, P. Grosch's law revisited: Cpu Power and the Cost of Computing. Communication of the ACM. 28(2), 142-150, 1985.

[9] Faye Borthick, A. and Roth, H. P. Understanding Client/server Computing. Management accounting, 36-41, aug. 1994..

[10] Gavish, B. and Pirkul, H. Computer and Database Location in Distributed Computer Systems. IEEE trans. on Computers. 35(7), 583-590, 1986.

[11] Grosch, H. A. High Speed Arithmetic: the Digital Computer as a Research Tool. Journal of Operational Research. 43(4), 1959.

[12] HP. The scope of HP Systems Consolidation. http://www.hp.com/products1/unixservers/solutions/sc/scope.html, 2002.

[13] IBM. Improve the return on your IT Investment with Server Consolidation. http://www-1.ibm.com/servers/eserver/iseries/australia/serv.htm, 2002.

[14] Jain, H. K. A Comprehensive Model for the Design of Distributed Computer Systems. IEEE Transactions on software engineering. 13(10), 1092-1104, 1987.

[15] KPMG consulting. Read your efuture, 2000.

[16] Lengauer, T. Combinatorial Algorithms for Integrated Circuit layout, John Wiley & sons, Chichester, 1990.

[17] Liu, C. and Layland, J. Scheduling algorithms for Multiprogramming in a hard real-time Environment. Journal of the ACM. 30(1), 46-61, 1973.

[18] Menascé D. A., Gomaa, H., Kerschberg, L. A Performance Oriented Design Methodology for large-scale Distributed Data Intensive Information Systems. IEEE 1st International conference on engineering of complex computer systems, 1995.

[19] Menascé, D. A. and Almeida, V. A. F. Scaling for e-business. Technologies, Models, Performance and Capacity Planning. Prentice-Hall, 2000.

[20] Microsoft. 2000. Introducing Windows 2000 clustering technologies. http://www.microsoft.com/windows2000/techinfo/howitworks/cluster/introcluster.asp

[21] Moad, J. Client/server costs: don't get taker for a ride. datamation. 15(2), 34-41, 1994.

[22] Molta, D. For client/server, think thin. Network Computing. www.networkcomputing.com/1013/1013f1.html, 1999.

[23] Molta, D. Thin client computers come of age. Network Computing, 1999.

[24] Papadimitriou, C., Steiglitz K. Combinatorial Optimization. Prentice Hall, inc., Englewood Cliffs, New Jersey, 1982.

[25] Robert Frances Group. Server Consolidation www.ibm.com/servers/solutions/serverconsolidation/pdf/robertfrancesgroup.pdf, 1999.

[26] Schlegel, K. PC Prices falling as Percentage of TCO. Meta Group. www.metagroup.com, 2002.

[27] Stevens, L. Strategies for Distributed Systems. http://www.uniforum.org/news/html/publications/ufm/jan97/thin.html, 1997.

[28] The Tolly Group. Total cost of Application Ownership. www.tolly.com, 1999.

[29] Umar, A. Object-oriented Client-server Internet environments. Prentice-Hall, 1999.

[30] Willcocks, L. Evaluating Information Technology investments: research, findings, and reappraisal. Journal of Information Systems. 2, 243-268, 1992.

[31] Wolf, J., Yu, P. S. On balancing the load in a Clustered Web farm. ACM transactions on Internet technology. 1(2), 231-261, 2001.

Danilo Ardagna is a Ph.D. student at Politecnico di Milano, where he also graduated in Computer Engineering in December 2000. He has worked for a 6-month period at IBM T.J. Watson Research Center in the System Optimization Department. His research interests include IT infrastructural design and cost minimization, capacity planning and scalability.

Chiara Francalanci is an associate professor of Information Systems at Politecnico di Milano. She has a master's degree in Electronic Engineering from Politecnico di Milano, where she has also completed her Ph.D. in Computer Science. As part of her post doctoral studies, she has worked for two years at the Harvard Business School as a Visiting Researcher. She has authored articles on the economics of information technology and on feasibility analyses of IT projects, consulted in the financial industry, both in Europe and the U.S., and is a member of the editorial board of the Journal of Information Technology.

Marco Trubian is an associate professor of Operations Research at the Computer Science Department of the Universìtà degli Studi of Milan. He has taken his Ph.D. degree in Electronic and Systems Engineering at the Dipartimento di Elettronica e Informazione of Politecnico of Milan in 1992. His research activities are in the area of combinatorial optimization and are mainly devoted to the modeling and algorithmic aspects. The main investigated application fields are those related with scheduling and telecommunication problems. An uninterrupted research field is the design and the development of efficient heuristic algorithms based on local search techniques. In the same time he has developed exact algorithms for polynomially solvable and NP-hard problems. He is referee of several international journals.