# COST-EFFECTIVE SERVICE TASK ASSIGNMENT IN MULTI-DOMAIN DISTRIBUTED COMPUTING ENVIRONMENTS

Malamati Louta, Angelos Michalas, Ioannis Psoroulas
*School of Electrical and Computer Engineering, National Technical University of Athens, 9 Heroon Polytechneiou Str, Athens, Greece*

Evangelos Loutas
*Department of Informatics, University of Thessaloniki, Thessaloniki, Greece*

Keywords: Intelligent Agents, Service Task Assignment, Service Nodes, Domains, Resource Allocation

Abstract: Highly competitive and open environments should encompass mechanisms that will assist service providers in accounting for their interests, i.e., offering at a given period of time adequate quality services in a cost efficient manner which is highly associated to efficiently managing and fulfilling current user requests. In this paper, service task assignment problem is addressed from one of the possible theoretical perspectives, while new functionality is introduced in service architectures that run in open environments in order to support the proposed solution. The pertinent problem aiming to find the most appropriate assignment of service tasks to service nodes is concisely defined, mathematically formulated and empirically evaluated.

## 1 INTRODUCTION

The ongoing liberalisation and deregulation of the telecommunication market will introduce new actors. In principle, the main role of all players in such a competitive environment will be to constantly monitor the user demand, and in response to create, promote and provide the desired services and service features. According to a business model foreseen to apply to the telecommunications world of the near future, five main different entities can be identified, namely, user, service provider, third party application or/and content provider, broker and network provider. The role of the third party application or/and content provider is to develop and offer applications or/and content. The role of the service provider is to provide the means through which the users will be enabled to access the applications or/and content of third party application or/and content providers. The broker assists business level entities in finding other business entities. Finally, the role of a network provider is to offer the network connectivity needed for service provision.

Service provisioning in such open models is a quite complex process since it involves various diverse actors. The following are some key factors for success. First, the efficiency with which services will be developed. Second, the quality level, in relation with the corresponding cost, of new services. Third, the efficiency with which the services will be operated, controlled, maintained, administered, etc. The challenges outlined above have brought to the foreground several new important research areas. Some of them are the elaboration on e-business concepts (Ghosh, 1998), the specification of service architectures (SAs) (Magedanz, 1997), the development of advanced service creation environments (SCEs) (Tag, 1996) and service features (e.g. the personal mobility concept), and the exploitation of advanced software technologies, (e.g. distributed object computing (Vinoski, 1997) and intelligent mobile agents (Morreale, 1998). The aim of this paper is, in accordance with the cost-effective QoS provision and the efficient service operation objectives, to propose enhancements to the sophistication of the functionality that can be offered by service architectures in open competitive communications environments.

In accordance with the SA concept and exploiting advanced software paradigms, the service logic is



Louta M., Michalas A., Psoroulas I. and Loutas E. (2005).
COST-EFFECTIVE SERVICE TASK ASSIGNMENT IN MULTI- DOMAIN DISTRIBUTED COMPUTING ENVIRONMENTS.
In *Proceedings of the Second International Conference on e-Business and Telecommunication Networks*, pages 164-168
DOI: 10.5220/0001407901640168

realized by a set of autonomous co-operating components, which interact through middleware functionality that runs over Distributed Processing management and resilience/ redundancy purposes it is assumed that each service provider deploys service components realising service logic in different service nodes, residing in the same and/or different domains. In the context of this paper domains represent different network segments. Moreover, it can be envisaged that a service will in general comprise a set of distinct service tasks, which could be executed by different service nodes.

Highly competitive and open environments should encompass mechanisms that will assist service providers in accounting for their interests, i.e., offering at a given period of time adequate quality services in a cost efficient manner which is highly associated to efficiently managing and fulfilling current user requests. Assume that a user wishes to access a specific service offered by a service provider. The service is composed by a distinct set of service tasks. Each service task can be served by various candidate service nodes (CSNs). Thus, a problem that should be addressed is the assignment of service tasks to the most appropriate service nodes. In this paper, the pertinent problem is called service task assignment. The aim of this paper is to address the problem from one of the possible theoretical perspectives and to show the software architecture that supports its solution

This study is related to pertinent previous work in the literature. Efficient resource utilisation and job scheduling is gaining the attention of the researchers as computational grids (interconnected networks of super-computing centers) have become an emerging trend on high performance computing (Special Issue, 2003). Most studies in the field of resource allocation schemes aim at efficiently utillising the CPU resources spread throughout a network. Different global objectives could be considered, such as minimization of mean service/task completion time, maximization of resources utilization (e.g., CPU utilization), minimization of mean response ratio (Tanenbaum, 2001). The design choices that the system architect has to face are quite vast ranging from deploying centralized vs. decentralized arrival and/or control systems, adopting static vs. dynamic schemes, considering different resource allocation strategies/algorithms incorporating or not the task migration concept, taking into account diverse load metrics, etc. (Stone, 1997). In general, many approaches have derived and encourage the necessity of adaptive switching between strategies and dynamic adjustment of Environments (e.g., Common Object Request Broker Architecture - CORBA). Limited by techno-economic reasons or considering administrative, decision parameters (e.g., node's load predefined threshold, time interval upon which load information exchange between the nodes should take place).

The approach in this paper is the following. Section 2 provides the software elements required for the realization of the service task assignment process, while our assumptions regarding the system model are presented. Section 3 presents a concise definition and mathematical formulation to the service task assignment problem. Finally, section 4 gives future plans and concluding remarks.

# 2 SOFTWARE ARCHITECTURE

Service Architectures (e.g., TINA Access Session (Magedanz, 1997)) offer the framework for user authentication and service invocation. The feature that is not supported is the overall task of service task assignment. The following key extensions are made so as to cover the necessary functionality. First, the Service Provider Agent (SPA) is introduced and assigned with the role of selecting on behalf of the service provider the best service task assignment pattern. Second, the User Agent (UA) is assigned with the role of promoting the service request to the appropriate SPA. Third, the Service Node Agent (SNA) is introduced and assigned with the role of promoting the current load conditions of a CSN. Finally, the Network Provider Agent (NPA) is introduced and assigned with the task of providing current network load conditions (i.e., bandwidth availability) to the appropriate SPA. In other words, the SPA interacts with the UA in order to acquire the user preferences, requirements and constraints, analyses the user request in order to identify the service tasks constituting the service and their respective requirements in terms of CPU, memory and disk space, identifies the set of CSNs and their respective capabilities, interacts with the SNAs of the candidate service nodes so as to obtain their current load conditions and with the NPAs so as to acquire the network load conditions, and ultimately selects the most appropriate service task assignment pattern for the provision of the desired service.

Regarding the system model, we consider a set of service nodes $SN$ and a set of links $L$. Each service node $n_i \in SN$ corresponds to a server, while each link $l \in L$ corresponds to a physical link that interconnects two nodes $n_i, n_j \in SN$. Our system

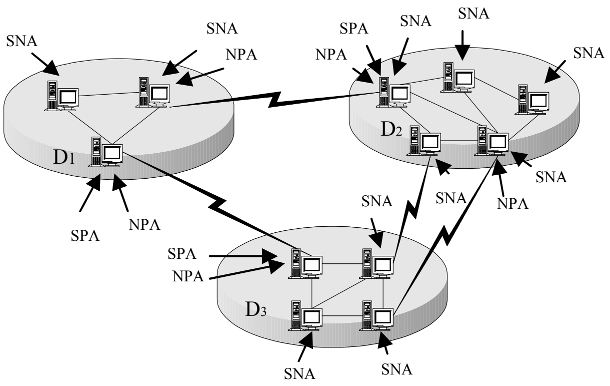


Figure 1: System Model and physical distribution of the service task assignment related components.

operates in a multi-tasking environment, i.e., several tasks may be executed on a single service node sharing its resources (e.g., CPU utilization, memory, disk space). Let $D_i$ denote a set of nodes grouped to form a domain. A pattern for the physical distribution of the related components to the service task assignment scheme is given in Figure 1. Each SPA controls the service nodes of a domain. Each SNA is associated with each node, while each NPA is associated with the network elements (e.g., switches or routers) necessary for supporting service node connectivity. The SNA, NPA role (in a sense) is to represent the service nodes or network elements, respectively, and to assist SPA by providing information on the availability of resources of the service node/network element. Domain state information (load conditions of the service nodes of the particular domain and link utilisation) is exchanged between the SPA and the SNAs/NPAs residing in the specific domain, while SPAs residing in different domains exchange their domain state info. This approach increases scalability as it reduces the requirements in terms of computation, communication and storage. At this point it should be noted that for simplicity reasons the network elements needed for the service node connectivity are not depicted in Figure 1.

In the scope of this paper we consider that the service nodes constituting a specific domain are interconnected by a local area network, while different domains are interconnected by a wide area network. In the current version of this study we limit our attention to the cases where a service request may be served by service nodes residing in a single domain, since we consider that the cost imposed due to information transfer through the WAN links is big, diminishing the net benefit of possible efficient resource utilisation. Thus, in our study, in case a service request cannot be served by the service nodes of a domain, it is transferred to the SPA of another domain in order to handle the request. However, the formulation of the service task assignment problem is given in a general mode, since the emergence of infrastructure (e.g., very high performance Backbone Network Service (vBNS) , NSFnet topologies) and test-beds like Tera-Grid (TeraGrid, 2003) promises remarkable network bandwidth between distant sites, enabling thus load balancing with minimal cost.

# 3 PROBLEM FORMULATION

User $u$ wishes to use a given service $s$ , which may be decomposed in a set of distinct service tasks, which will be denoted as $ST(s)$. Among these service features, those of interest to the user will be denoted as $ST(u,s)$, where $ST(u,s) \subseteq ST(s)$.

Let's assume the existence of multiple service nodes for the provision of service $s$, denoted by $SN(s) = \{n_1,...,n_{|k|}\}$. Each service node-$n_j$ contains a collection of components, denoted as $A_{n_j}(i)$, which inter-work with other components that may reside in the same or in a different service node in order to accomplish each service task $i \in ST(s)$. Let $A_{n_j}$ and $C$ be the total set of components residing in the $n_j$ service node and the various service nodes in total, respectively. Hence, the following relationship holds: $A_{n_j}(i) \subseteq A_{n_j} \subseteq C$. Each service task $i \in ST(s)$ may be executed on an associated set of possible candidate service nodes, represented by the set $SN(i)$, ($i \in ST(u,s)$). Thus, $SN(i) \subseteq SN(s)$. The service logic deployment pattern adopted by service providers determine each of these service node sets.

Task $i$, ($i \in ST(s)$) requires for its completion consumption of $r_{CPU}(i)$, $r_{mem}(i)$ and $r_{disk}(i)$ resources of service node(s) $n_j, (n_j \in SN(i))$. A realistic assumption is that SPA being in charge of assisting the service providers in the competitive telecommunication market, has a solid interest in as accurately as possible identifying the resources $r_a(i)$ (where $a \in \{CPU, mem, disk\}$) needed for the provisioning of service task $i$ in terms of CPU utilization, memory and disk space. In this respect, the SPA can be the entity that configures these values based on the service feature characteristics, user preferences and requirements, exploiting also previous experience.

Let $c_D$ denote the cost of involving service node $n_j, (n_j \in SN(i))$, in the service provision. For notation simplicity it is assumed that the cost of involving a service node in the solution is the same for all service nodes. Notation may readily be extended.

The objective of our problem is to find a service task assignment pattern, i.e., an assignment $A_{ST}(s)$ of service tasks $i$ ($i \in ST(u,s)$) to service nodes $n_j, (n_j \in SN(i))$, that is optimal given the current load conditions and number of service tasks being served by each service node $n_j$, represented as $r_a^{pre}(n_j)$ and $k^{pre}(n_j)$, respectively. The assignment should minimise an objective function $f(s, A_{ST}(s))$ that models the overall cost introduced due to network resources consumption. Among the terms of this function there can be the overall cost due to the deployment of various service nodes to the service provisioning process, the communication cost introduced due to the interaction of the components $A_{n_j}$ residing in $n_j$ service node with the components $A_{n_k}$ residing in service node $n_k$ for the completion of each service task $i$, $(\forall i \in ST(s))$, as well as the management cost $c_M(i,i')$ introduced due to the assignment of $(i,i') \in ST^2(s)$ service tasks to different service nodes $(n_j, n_{j'}) \in SN^2(i)$.

The constraints of our problem are the following. First, each service task $i$ ($i \in ST(u,s)$) should be assigned to only one service node $n_j$, $(n_j \in SN(i))$. Second, the capacity constraints of each service node should be preserved. Lets assume that $r_a^{\max}$ and $k^{\max}$ represent the maximum load and the maximum number of tasks that a service node may handle. For notation simplicity, these parameters are assumed to be the same for each service node $n_j$, $(n_j \in SN(i))$. Thus, the constraints are $r_a^{post}(n_j) \le r_a^{\max}$ and $k^{post}(n_j) \le k^{\max}$, $(\forall n_j \in SN(i))$, where $r_a^{post}(n_j)$ and $k^{post}(n_j)$ denote the potential load conditions of service node $n_j$, after the service task assignment process. Notation may readily be extended.

In order to describe the allocation $A_{ST}(s)$ of service tasks to service nodes we introduce the decision variables $x_{ST}(i,j)$ ($i \in ST(u,s)$, $n_j \in SN(i)$) that take the value 1(0) depending on whether service task $i$ is (is not) executed by service node-$n_j$. The decision variables $y_{SN}(j)$ assume the value 1(0) depending on whether candidate service node $n_j$ ($n_j \in SN(i)$) is (is not) deployed (involved in the solution). In addition, we define the set of variables $z_{ST}(i,i')$ ($\forall (i,i') \in ST(u,s)^2$) that take the value 1(0) depending on whether the service tasks $i$ and $i'$ are (are not) assigned to the same service node. Allocation $A_{ST}(s)$ may be obtained by reduction to the following 0-1 linear programming problem.

Service Task Assignment Problem:

Minimise

$$f(s, A_{TN}(s)) = c_D \cdot \sum_{n_j \in SN(s)} y_{SN}(j) \cdot (1 + b \cdot \sum_{a \in \{CPU, memory, disk\}} w_a \cdot \frac{r_a^{pre}(n_j)}{r_a^{\max}(n_j)}) + \sum_{i \in ST(s)} \sum_{n_j \in SN(i)} C(i,n_j) \cdot x_{ST}(i,j) + \sum_{i \in ST(s)} \sum_{i' \in ST(s)} c_M(i,i') \cdot (1 - z_{ST}(i,i')) \quad (1),$$

where $C(i,n_j)$ denotes the communication cost introduced in case $n_j$ service node has undertaken the responsibility for the execution of service task $i$ ($i \in ST(u,s)$),

subject to the constraints:

$$\sum_{n_j \in SN(i)} x_{ST}(i,j) = 1, \quad \forall i \in ST(s) \quad (2),$$

$$r_a^{pre}(n_j) + \sum_{i \in ST(s)} r_a(i) \cdot x_{ST}(i,j) \le r_a^{\max}(j) \cdot y_{SN}(j), \quad \forall n_j \in SN(s) \quad (3),$$

$$k^{pre}(n_j) + \sum_{i \in ST(s)} x_{ST}(i,j) \le k^{\max}(j) \cdot y_{SN}(j), \quad \forall n_j \in SN(s) \quad (4)$$

At this point it should be noted that, in order for the service providers to better utilize their resources, the cost of each service node deployment introduced in cost function (1) takes also into account the node's current load conditions in order to obtain a load balancing solution. Parameters $b$, $(b<1)$, and $w_a$ denote the relative significance of load balancing and of each resource type $a$ to the service provider. Constraints (2), guarantee that each service task will be assigned to one service node. Constraints (3) and (4) guarantee that each service node will not have to cope with more load and service tasks than those dictated by its pertinent capacity constraint.

# 4 CONCLUSIONS

This paper presented a mechanism that will assist service providers in efficiently managing and fulfilling user requests. Specifically, the contribution of this paper lies in the following areas. First, the definition and mathematical formulation of (one possible version) of the service task assignment problem. Second, the presentation of a software architecture, which is adopted for acquiring the best service task configuration pattern, i.e., assignment of service tasks to service nodes for efficient service provisioning.

Initial results indicate that the proposed framework produces good results in simple contexts where only the service node deployment cost factor is considered. Directions for future work include, but are not limited to the following. First, the realisation of further wide scale trials, so as to experiment with the applicability of the framework presented herewith. Second, the experimentation with alternate approaches (e.g., market-based techniques) for solving the service task assignment problem.

# REFERENCES

Ghosh S., 1998. “Making business sense of the Internet”, *Harvard Business Review*, vol. 76, no. 2, pp. 127-135.

Magedanz, T., 1997. “TINA-Architectural basis for future telecommunications services”, *Computer Communications*, vol. 20, no. 4, pp. 233-245.

Morreale P., 1998. “Agents on the move”, *IEEE Spectrum*, vol. 35, no. 4, pp. 34-41.

Special Issue, 2003. “Special section on grid computing”, ACM SIGMETRICS Performance Evaluation Review, vol. 30, no. 4, pp. 12-49.

Stone H., 1997. “Multiprocessor Scheduling with the Aid of Network Flow Algorithms”, *IEEE Transactions on Software Engineering*, vol. 3, no. 1, pp. 85-93.

Tag M., 1996. “Service creation environment engineering”, *Proc. Interworking’96 Conference*, Japan.

Tanenbaum A. S., 2001. *Modern Operating Systems*, Englewood Cliffs, New Jersey: Prentice-Hall, 2nd ed.

The TeraGrid Project, 2003. *A distributed computing infrastructure for scientific research*.

Vinoski S., 1997. “CORBA: Integrating diverse applications within distributed heterogeneous environments”, *IEEE Commun. Mag.*, vol. 35, no. 2, pp. 46-55.