AN INTRA- AND INTER-DOMAIN PLACEMENT USING A MULTICRITERIA METHOD TO OPTIMIZE RESOURCE ACCESS IN CORBA

Huah Yong Chan and Bénédicte Herrmann

Laboratoire d'Informatique de Besançon
16 route de Gray 25030 Besançon France
{chan | herrmann} @comte.univ-fcomte.fr

Abstract: The aim of our study is to enhance the performance of distributed object applications, by positioning components in a manner that will distribute resource loads of mainly CPUs and networks. For this purpose, we suggest a multicriteria approach. We also present an architecture and a mechanism for the intra- and inter-domain object migration which is implemented in a CORBA compliant system.

Keywords: Multicriteria, domain, placement, optimization

1 INTRODUCTION

Development of distributed object applications such as electronic commerce, web applications increases in great proportions since the last few years. In this context, optimizing the execution of distributed applications in large networks becomes an important issue.

In this paper we evaluate a multicriteria decision procedure in placing objects in a CORBA compliant platform. At first, we define the context of our work, then we will present multicriteria decision in object placement decision. We also introduce a framework for resource allocation of intra- and inter-domain. Finally, we will describe our implementation of intra- and inter-domain placement and the test applications.

2 THE CONTEXT

2.1 Related work

In [4], processor allocation is based on processor load, completed by co- or counter-allocation indicators. Two objects with a co-allocation indicator will always be placed on the same node; if one of them is migrated, the other will follow but it is opposite for the conter-allocation indicator. The Computational Field Model (CFM) [8] proposes to find an optimal placement by computing balance calculus on a set of objects subjected to a set of forces. Each forces result from object computations and communications.

The study we made [3] concerns distributed applications implemented in CORBA-compliant distributed objects system. Distributed applications are made up of components. Each component is dedicated to a specific part of the global application goal. A complex distributed application can be composed of a lot of components.

A component consumes an amount of processor time, memory space and other resources. This consumption is hard to predict and thus has to be observed dynamically. The two major resources consumed by a component are processor load and network bandwidth. But optimizing these two resources is hard because they are opposed to each other. To optimize processor load, we need to distribute the load on all nodes on the network. However, to optimize network load, we need to group all communicating components on a same node.

Multicriteria aggregation procedures are well studied in [5]. We use aggregation leading to a single criterion in our decision procedure. Using this method, we have to normalize each criterion which means that converting the different criteria into the same unit so that we can compare them to each other.

2.2 Intra- and inter-domains

A domain [7] is a distinct scope, within which certain common characteristics are exhibited and common rules observed. We can define domains based on two criteria, one is technological and the other administrative.

Our domain is based on homogenous machines connected within the same local area network. Migrating to different domains poses more problems compared to migration within the same domain. This is because inter-domain uses unreliable networks (wide area networks) which faces problem of security and heterogeneity. Especially in inter-domain communications, we can not guarantee the honesty and the good use of users. Thus the issue of security is extremely important when we deal with migration. In order to solve this problem, we introduce contracts which are used to establish the relationship between domains, and inside these contracts one can specify the access and migration rights of objects. These contracts add one more security control besides existing security features such as encryption and authentication.

The migration can not be done if the compatible binary code for the architecture of target machine in the different domain does not exist. Even in the same class of machine, we can have problems of heterogeneity which are less obvious such as the resources (memory, disc space, printer) available in the machine or the various services (different quality of service, different version) provided by the machine. This heterogeneity imposes the constraints for the migration, especially for the applications that need specific resources.

In order to extend our model to multiple heterogeneous domains, we propose an architecture for the management of intra- and inter-domain migration [1].

3 DYNAMIC PLACEMENT USING MULTICRITERIA ANALYSIS

For this short paper, only a brief description of our method can be given here. See [3] for a complete description. To use multicriteria analysis in dynamic placement, we observe the system from at least two criteria : processor load and relation between objects. To elaborate a criterion from a point of view, it is necessary to provide a function on information given by the system which characterize consequences, in terms of performances, of moving an object.

A utility function is used to combined these two criteria, and an addition approach is chosen. We attach a significance factor to each criterion. This value determines the importance of one criterion compared to the others. Veto is also used for each criterion, to assure that minimum requirement for each criterion is respected. Different values of significance factors and vetos are evaluated in our tests.

4 AN ADMINISTRATION OF INTRA- AND INTER-DOMAIN APPLICATION

We have extended the first implementation of a dynamic object positioning server described in [2]. The earlier version can allow only intra-domain migration. The latest allows inter-domain migration and management of object constraints. It is actually composed of six modules. There are three information services, one decision service, one contract service and one domain authority service. There are two layers, namely site and domain. Intra-domain placement involves only the sites within the same domain, while inter-domain placement can cross the boundary to other domains.

load information module is in charge of maintaining up-to-date load information and informing the decision module when specific events occur, for instance big load imbalance. Load module gathers information from other sites and compute a local load indicator. By using the bidding algorithm, load indices are exchanged between load modules to evaluate global load on the network and to find light loaded sites. We use the fixed length list which keeps the most interacting nodes and one random node within the same domain to reduce the traffic. Thus we only exchange the load information with the nodes in this list. Each load module has their own list of nodes. The most interacting nodes are dynamically changed and the change depends on the consultation of local decision module. We use the receiver-initiated approach to inform heavy loaded nodes when the local site is idle and the difference of load between them is big enough.

relation graph module is in charge of maintaining up-to-date relation information and to analyze them. Each time a local object communicates, it informs the relation graph module. The relation graph uses a knowledge drift model [6] to identify the change of relationships between objects. Our relation graph maintains a list of local server objects and each local object has a list of distant sites and distant domains which have the relationships with it. When the relational part of an object O to a site S (ratio of communication to S and total communication of O) grows and becomes more important, then moving O to S is considered as a possible action. By observing this, the relation graph module selects possible actions and proposes them to the decision module.

constraint module is in charge of maintaining a per-object constraint list and a per-domain property list. By matching these two lists, the module can evaluate the constraint criterion. It can also propose to the decision module object which have dynamically changed their constraints and need to be moved. For current implemetation, this constraint module is used only for inter-domain migration and the constraints that we consider are hardware architecture, operating system, bandwidth of networks and security level.

decision module is in charge of evaluating actions, making decisions and positioning objects. It asks periodically the relation module and load module for the best of the possible actions by using the multicriteria analysis mentioned above. It also can evaluate actions proposed by the other modules when an evaluation is ask for. Current implementation just deals with object migration but we can extend our model to include other decision modules which deal with object replication and aggregation. We also consider the communication criterion only for inter-domain migration and make sure the constraints of objects are respected.

contract module updates and keeps the contracts between domains. By using contract, we can describe which domains to be federated and what sort of service to be shared.

domain authority is responsible in establishing relationships between domains by using contracts and it verifies the access and the placement rights for inter-domain. It acts as a representative vis-a-vis the other domains.

5 IMPLEMENTATION

The load information module, the relation graph module and the decision module are distributed in each site of the domain. For the domain level, the contract module, the constraint module and the domain authority exist in each domain.

Our object migration is done using lifecycle services defined by OMG. We use the factory to create the object at where we want to move to. In this context, the object here refers to the servant (instant of CORBA object) attached to the Object Adapter.

For the object migration, we can use two strategies within the same domain. The first strategy, it is based on our relation graph to identity the candidate object and target node for migration, then get the load index of target node from load module. The evaluation is made in decision module based on the multicriteria analysis to determine whether it is worth to migrate. The second strategy, it is based on the bidding algorithm. When an underloaded node is found, it will identity the source node which has the highest bid (the most loaded node) in its list and inform the source load module. The source load module will ask the source decision module for an evaluation. The decision module consults the source relation graph module for the candidate object (the least communication object with the source node), then an evaluation is made.

In the case of inter-domain object migration, we use the following strategy. The decision module evaluates the proposition of the relation graph module whether distant communication intensity of candidate object is greater than its local communication intensity plus veto of communication. If the evaluation is good then the decision module will ask the source domain authority to verify from the target domain authority whether the migration can be carried out on respecting the contracts and the constraints of the object. The reason for not taking the computation load index into considerion is to minimize the overhead of traffic accross the network, and our main objective of inter-domain migration is to improve the quality of service such as the availability and the security which can be specified in our constraints.

6 TESTS

We have carried out some empirical tests to do quantitative analysis and try to show the correlation between different parameters and the behavior of applications. Thus we have developed as well a simulator of various test applications. This simulator enable us to launch servers and clients in the domains and the sites that we specify and to create the relationships between clients and servers. We can determine the ratio of commuciation and computation for each server and client in our simulator, and the intensity of communication between clients and servers. The various tests are carried out within the same domain first. The set of parameters such as the load factor and the relation factor, delay periods, vetoes and thresholds are tested with different scenarios of test applications.

Our results showed that we have around 3 to 5% overhead. The benefits that we can gain depend on the scenario of applications and the environment of execution. For example if the volume communication between two different site objects are very high and the server and the client object consume very little CPU for computation, then our system will place the server object and the client object together in the same site, in order to optimize the communication criterion. By doing this, our tests showed the execution time of a client can be three times faster compared to without migration. This is only a simple scenario which shows the communication criterion become very important when the network is heavily used.

6.1 Inter-domain test scenario

The following test is to verify the qualitative analysis. Test has been carried out on 3 domains, refer to figure 1. The platform of the first two domains are sun workstations run on Solaris, and the third is Intel workstations run on Solaris. Each domain is connected by Ethernet and consists of four machines. In this test, we show that the placement can be carried out in different domains and on respecting the constraints that we imposed on the object. Even though the domain 3 has the most invocations to the server S, the server S is migrated to domain 1 because of the security constraint that we imposed on server S must be as good as or better than S2. This migration minimizes the communication between domain 1 and domain 2, and can have a very

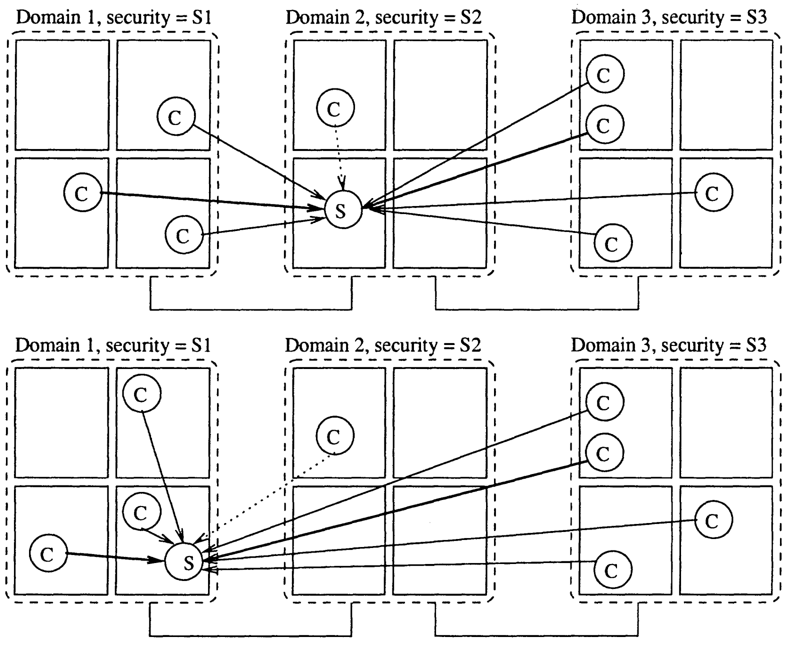

Figure 1. Inter-domain migration.

substantial gain if the communication between these domains are using a very slow and unreliable wide area network.

7 CONCLUSION

In this paper we have presented an automatic placement architecture designed for object based systems. We have focused on a multicriteria decision procedure used to aggregate several information. Our system can also scale well to wide area networks that consists of heterogeneous domains. The interest of dynamic object placement has been shown by our tests. Our implementation is also verified by a simulator of various test applications. We discover that the set of parameters chosen in the decision procedure depend on the behavior of applications. Hence an auto-adaptive approach to adjust the parameters is desirable for our future work.

References

[1] CHAN, H.Y., AND PHILIPPE, L., *Un gestionnarire de placement interdomaine sur CORBA*. RenPar'10, Strasbourg, France, June-1998.

[2] CHATONAY, P., BOURDON, F., HERRMANN, B., AND PHILIPPE, L., Dynamic Object Positioning. *Proceeding of ECOOP'96*, LNCS, July-1996.

[3] CHATONAY, P., *Management of resource allocation for objects in distributed systems, a multicriteria approach integrating communications*. Phd Thesis, Université de Franche-Comté, 1998.

[4] DICKMAN, P., Effective load balancing in a distributed object-support operating system. *Proc of the 2nd International Workshop on object orientation in OS*, IEEE, 147-153, 1992.

[5] ROY, B., BOUYSSOU, D., *Aide Multicritere à la décision : Méthodes et Cas*. Economica, 1992.

[6] BOURDON, F., The Automatic Positioning of Objects in COOL v2. *Proc of the 14th International Conference on Distributed Computing Systems (ICDCS)*, IEEE Computer Society Press, Poland, June, 1994.

[7] HOFFNER, Y., AND CRAWFORD, B., Using interception to create domains in distributed systems. *The joint international conference on Open Distributed Process ing & Distributed Platforms*, May, 1997.

[8] TOKORO, M., Computational Field Model: Toward a New Computing Model/Methodology for Open Distributed Environment. *Proc of the 2nd Workshop on future trends in distributed computing systems*. Egypt, September-1990.

Biographies

Huah Yong Chan is a Ph.D. student in Université de Franche-Comté. His domain of research is in resource allocation in distributed object system.

Bénédicte Herrmann is a lecturer in Université de Franche-Comté. Her domain of research is in load balancing and replication in distributed object system.