

Self-Organizing Resource Allocation for Autonomic Networks

T. Eymann, M. Reinicke

*Institute for Computer Science and Social Studies
Albert-Ludwigs-University Freiburg, Germany*

eymann, reinicke@iig.uni-freiburg.de

O. Ardaiz, P. Artigas, F. Freitag, L.
Navarro

*Department of Computer Architecture
Technical University of Catalonia, Spain
oardaiz,partigas,
felix,leandro@ac.upc.es*

Abstract

Application-layer networks (ALN) are software architectures that allow the provisioning of services requiring a huge amount of resources by connecting large numbers of individual computers, e.g. Grids and P2P-Networks. Self-organization, like proposed by the Autonomic Computing concept, might be the key to controlling these systems. The CATNET project evaluates a decentralized mechanism for resource allocation in ALN, based on the economic paradigm of the Catallaxy. The economic model is based on self-interested maximization of utility and self-interested cooperation between software agents, who buy and sell network services and resources to and from each other.

1. Allocation of Resources in Application Layer Networks

Application-layer networks (ALN) are software architectures that coordinate the provisioning of services requiring a huge amount of resources by connecting large numbers of individual computers. Global Internet-based networks, like today's Grids [1] and Peer-to-Peer-Computing [14], take advantage of such infrastructures with applications like multicast services for global audiences, storage repositories of peta-scale data sets, or parallel computing applications requiring teraflops of processing power.

Such applications are executed in multiple resource locations distributed throughout the Internet, coordinated on the application layer using a dedicated network, the ALN. An ALN scenario would be the distributed provisioning of web services for Adobe's Acrobat (for creating PDF files). Here, word-processor client programs would transparently address the nearest/ cheapest Acrobat service instance in order to create PDF files. The overall objective of the ALN would be (a) to always provide access to some Acrobat service instance, such that a minimum number of service demands have to be rejected, and (b) to optimize network parameters such as provisioning and transmission costs. This paper assumes

that the future development of these applications will lead to clients paying for the access to a service and the corresponding on- or offline exchange of payment; the individual goal of a client would become to access a service cheaply, while services may try to maximize income.

These ALN will work best if they include some of the properties ascribed to the upcoming Autonomic Computing (AC) paradigm. First of all, these systems possess a system identity, so that the single service provider knows whether it belongs to the system or not, and the client addresses its requests "to the system" rather than to a bunch of unconnected nodes. As the demand and offer situation in the network continuously and (assumably unpredictably) varies, the AC system reconfigures these resources such that the objectives of availability and resource usage are sufficiently met. On the application level, this variation might be caused by nodes entering or leaving the network (e.g. in P2P networks); on a technical level, network connections might disappear or get overly congested. The required adaptation and learning mechanisms to cope with these kind of situations contribute both to self-healing of the system and to continuous optimization of its processes.

Controlling these systems is principally achieved through application of service control and resource allocation mechanisms. Their basic purpose is to match service supply and demand, in the likely case of multiple, redundant service instances, to meet those objectives. The simple service discovery mechanisms available today in decentralized networks (e.g. Jini [19]) seldom provide such functionality, as the case of redundant service instances is yet rare. In principle, resource allocation can be either achieved *explicitly* (using a centralized optimization or coordination instance) or emergently (by coherent action of decentralized system elements). However, a centralized realization using a coordinator instance (auctioneer, arbitrator, dispatcher, scheduler, manager), like e.g. in GLOBUS [7] or CONDOR-G [8], has several drawbacks.

As service demands and nodes connectivity changes are frequent, and new different services are created and

composed continuously, snapshots taken of the system state are soon outdated. Any solution centrally computed on the basis of this information tries to optimize a past state of the network, which may be inconsistent to the current one. Dynamic ALN need a continuous, real-time coordination mechanism, which reflects the changes in the environment.

A second related property is that the coordinator needs to have global knowledge on the state of the network. This is mostly achieved by calculating the time steps such that actual status information from all nodes arrives safely at the coordination instance. However, if the diameter of the network grows, this approach leads to long latency times for the nodes.

Third, a centralized coordinator is part of the problem that decentralized ALN are trying to solve: As bids and offers have to route through the network to the single instance which collects global knowledge and computes the resource allocation, the distribution and deployment of services throughout the network is counteracted. This is currently not a problem as the control information is small compared to the allocation data itself, but may increase when the principle is applied to more and more application areas.

These drawbacks lead to the search for a truly decentralized coordination concept which is able to allocate services and resources in real-time without a dedicated coordinator instance. This concept should on one hand be able to cope with technical shortcomings like varying amounts of memory and disk space, internet connection speed and sporadic appearance and disappearance of the services. On the other hand, it is desirable that the network as a whole shows optimized behavior with regard to low overhead communication, short computation times, and economical resource allocation.

The Autonomic Computing paradigm is based on a biological notion of self-organization [11]. However, recent research in Grid computing has recognized the value of price generation and negotiation, and in general investigates economic models for trading resources and services and the regulation of supply and demand of resources in an increasingly large-scale and complex Grid environment [3].

In the remainder of this article, we first introduce a decentralized economic concept for coordination, the *Catallaxy*, and describe the *CATNET* project. The following section shows experiments conducted in the *CATNET* environment compares money and message flows in the application-layer network economic model, both with a centralized (baseline) and a decentralized implementation. Next we describe how the experiments are conducted in both cases. The article closes with some preliminary experimental results and an outlook to further research.

2. The *Catallaxy* Paradigm and the *CATNET* Project

The *Catallaxy* coordination approach [6] is an economic coordination mechanism for information systems consisting of autonomous network elements, based on constant negotiation and price signaling.

In Adam Smith's description of the "invisible hand" [16], market participants compete for limited resources and coordinate themselves through pursuance of their own interest. Hayek's *Catallaxy* [9] is a later and similarly decentralized concept of the neo-austrian school of economics, opposed to the centralized Walrasian auctioneer and Keynesian computable general equilibrium approaches. The term *Catallaxy* derives from the Greek word *katallatein*, which means "barter" and at the same time "to join a community". The central characteristics of *Catallaxy* are (1) agents working in their own interest to gain income, (2) who subjectively weigh and choose preferred alternatives, and (3) who communicate using commonly accessible markets. The subjective decision making is caused by "constitutional ignorance", which takes into account that it is (inevitably) impossible to know each and every environment detail that determines the agent's action.

However, a formal description of this self-organizing market mechanism does not so far exist. The research field of agent-based computational economics [18] tries both to provide formal descriptions and to build computer simulations. Starting from these simulations, the *Catallaxy* approach aims to build large multiagent information systems. Their software agents are able to adapt their heuristic strategies using machine learning mechanisms [17], and this constant revision of strategies leads to a co-evolution of software agent strategies, a stabilization of prices throughout the system and self-regulating coordination patterns [5]. The resulting patterns are comparable to those witnessed in human market negotiation experiments [13].

Earlier work in the context of computer science has used economic principles for resource allocation in operating systems, packet routing in computer networks, and load balancing in distributed computer systems [4; 10]. Most of these approaches rely on using a centralized auctioneer and the explicit calculation of an equilibrium price as a valid implementation of the mechanism. A successful implementation of the *Catallaxy* paradigm for a distributed resource allocation mechanism promises the advantage of a more flexible structure and inherent parallel processing compared to a centralized, auctioneer-based approach. This comparison can be done using both economical and technical criteria.

For the economic evaluation of the overall success of the control mechanism we use the "maximum social welfare utility" (SWF) criterion, which is the sum of all

individual utility function values of the participating nodes [15]. Every Client, Service Copy or Resource gains individual utility from buying lower or selling higher than the perceived market price. It can be enhanced by doing more transactions in the same time, but communication costs subtract from it. In total, SWF balances revenues and cost throughout the network. Increasing performance and decreasing communication in the whole network thus directly computes to relatively maximize social welfare utility. Other evaluation parameters are communication cost, allocation efficiency, network traffic and service access latency.

The goal of the CATNET project is thus to evaluate the Catalaxy paradigm for decentralized operation of application layer networks in comparison to a baseline centralized system. To achieve this, we have developed the CATNET ALN simulator, which allows to experimentally comparing two main resource allocation strategies: A centralized approach in which allocation decisions are taken centrally and a decentralized approach, where local agents negotiate resources using economic models.

The CATNET ALN simulator is implemented on top of the JAVASIM [2; 12] network simulator. It can be configured to simulate a specific ALN, such as a content distribution network or peer-to-peer network. Different agent types can be instantiated, namely clients, resource agents, and service agents. Network resources to be allocated encompass service access, bandwidth and storage. The simulation builds on a TCP/IP network model supported by JAVASIM. It describes the generic structure of a node (either an end host or a router) and the generic network components, which can both be used as base classes to implement protocols across various layers.

4 Experimental evaluation

With experiments we wish to measure if an ALN coordinated by the Catalaxy paradigm is able to successfully provide service to requesting clients. A second goal is to compare qualitatively the obtained results with the centrally coordinated baseline system.

4.1 Experimental setup

In our experiments we explore as design space of the system the node density and node dynamics of the network (Figure 1). First, we simulate the P2P network with different densities of the service and resource providing agents in a high dynamics environment (Figure 1 experiments 1A-C). Then, we simulate the high node density network in environments with different values of the dynamics (Figure 1 experiments 2A-C).

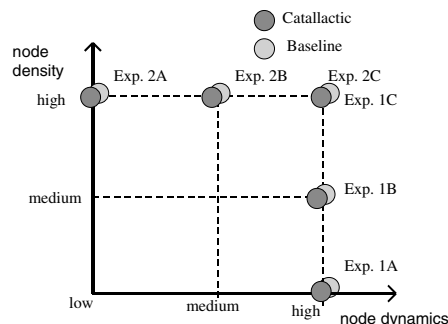


Figure 1. Design space of the experiments.

In the simulations the input is a trace of client demands with requests for service. The service request specifies the amount of service, a price, and its duration. In all experiments the same demand trace is used.

The physical network topology used in the experiments is organized in three levels of pentagons with leaves on the outer level. Although other specific or random topologies of the nodes could be used as well, we applied this topology since it facilitates controlled experiments.

On the top of the physical network an application layer network is build. Network nodes are instantiated having one of the previously described types of personalities, which can be a client, service copy or resource agent. Depending on the particular experiment, a node may contain several agents or no agent at all. In the second case, the node acts as a router.

The relation of the experimental configuration to real world ALN systems can be shown using P2P networks: A high value for the dynamics is interpreted to reflect the high level of connection and disconnection in P2P networks. A high level of node density represents the large number of machines with limited capability as found in P2P networks. In the simulations with high node density, we reduce the capacity of the resource agents, in order to represent small machines at the edge of the network. In the low node density scenario, on the other hand, the capacity of the service copy is increased such that the total amount of service available by the network is equal over all experiments.

Table 1 details the configuration of the experiments. The main parameters we are interested to measure are the number of client requests the network is able to provide a service for in the different scenarios. The scenario we are particularly interested in is the one with high node dynamics and high node density, as this configuration can be related to the conditions found in P2P networks.

Table 1. Experiment description.

Input trace	- 500 service request generated randomly by 75 clients over a time interval of 100 s. - each request is for 2 service units. - each service has a duration of 5 s.
Topology	- 106 physical nodes
Node density	- 75 clients on the leaves of the physical network - different density of resource and service copy agents. Each Resource has one service copy associated. Exp 1A: low node density: 5 resources with capacity 60. Exp 1B: medium node density: 25 resources with capacity 12. Exp 1C: high node density: 75 resources with capacity 4.
Node dynamics	Dynamic behavior: On average 70% of the service copies are connected. Exp 2A: Service copies do not change its state (static network) Exp 2B: Each 200 ms every service copy can change its state (connected/disconnected) with a probability of 0.2. Exp 2C: Each 200 ms every service copy can change its state (connected/disconnected) with a probability of 0.4.

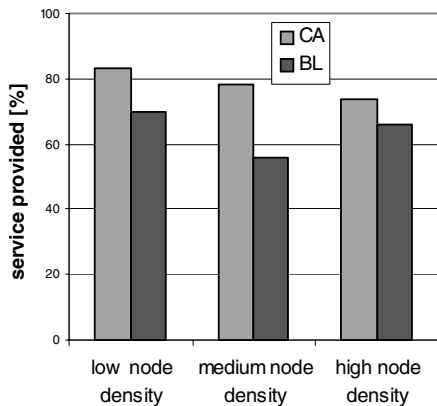


Figure 2. Service provision in % in a highly dynamic network environment with different node density.

4.2 Experimental results

In Figure 2 the service provision rate of a network with *different node density* in a highly dynamic environment is shown (experiments 1A – C). It can be observed that the network using catalactic coordination achieves a higher

service provision rates than the baseline system with a smooth decrease for increasing node density. In Figure 3 the service provision rate of a network with high node density in *different dynamic environment* is shown (experiments 2A – C). It can be observed that the service provision rate of the catalactic system is rather independent to the dynamics. The baseline system, on the other hand, decreases with increasing dynamics.

Considering the achieved service provision rate, our experimental results indicate that service provision in networks with many small nodes in a highly dynamics environment could be coordinated successfully by the catalactic paradigm.

Exploring additional parameters of the system has the potential to provide more insight in the behavior of such a complex system. Currently, we examine the influence of other parameters on the performance tendencies discovered so far. One of the drawbacks we found of the catalactic approach is the time needed to establish a service provision, which is high due to the negotiation protocol carried out by agents. Other parameters we study are e.g. how scale affects the performance of the system.

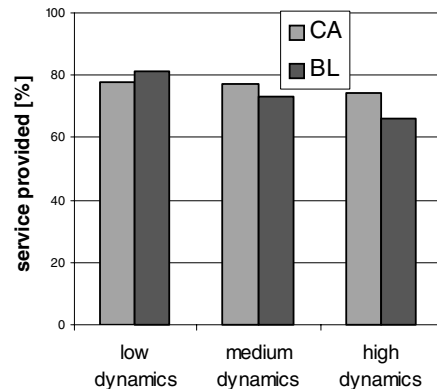


Figure 3. Service provision in % in a high node density environment with different dynamics.

5. Conclusion and Outlook

CATNET is a network simulator for ALN which can evaluate different resource allocation models. With regard to the Autonomic Computing concept, the evaluation of a self-organizing model against a centralized model is of particular interest. We have proposed the Catalactic coordination as a decentralized economic approach for resource allocation in AL networks. In this approach, the decisions of the agents are based on economic principles, being aware that resources like bandwidth, processing power and storage are limited.

Our simulations show that the service provision in the catalactic coordination is rather independent of the

dynamics of the network. The service provision capability of the baseline approach appears to be sensitive to the dynamics, reducing its performance in highly dynamic environments. These results not only show that self-organization might be beneficial for highly-dynamic systems. Because the Catallactic coordination approach is based on an economic paradigm, in contrast to the biological foundation commonly used for Autonomic Computing, it allows cost-consciously deciding from where a particular service should be retrieved and on which path it should travel. As an implicit benefit, this mechanism reduces the overall cost to provide the functionality that the AC paradigm proposes.

Acknowledgements

CATNET is supported by the European Commission Information Society Technologies Programme under contract no. IST-2001-34030.

References

- [1] D.P. Anderson and J. Kubiawicz, "The Worldwide Computer". *Scientific American*, Vol. 286, No. 3 (2002), pp. 28-35.
- [2] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, and H. Yu, "Advances in Network Simulation". *IEEE Computer*, Vol. 33, No. 5 (2002), pp. 59-67.
- [3] Buyya, R., *Economic-based Distributed Resource Management and Scheduling for Grid Computing*. Ph.D. Thesis. Monash University, Melbourne, Australia, 2002. <http://www.buyya.com/thesis/thesis.pdf>.
- [4] Clearwater, S.H. *Market-based control. A paradigm for distributed resource allocation*. World Scientific, Singapore, 1996.
- [5] Eymann, T., "Co-Evolution of Bargaining Strategies in a Decentralized Multi-Agent System". *AAAI Fall 2001 Symposium on Negotiation Methods for Autonomous Cooperative Systems*. AAAI, Falmouth, MA, 2001. p. 126-134.
- [6] Eymann, T., B. Padovan, and D. Schoder, "The Catallaxy as a new Paradigm for the Design of Information Systems". *Proceedings of The 16th World Computer Congress 2000 of the International Federation for Information Processing*. Beijing, China, 2000. <http://citeseer.nj.nec.com/eymann00catallaxy.html>.
- [7] I. Foster and C. Kesselman, "Globus: A Metacomputing Infrastructure Toolkit". *International Journal of Supercomputing Applications*, Vol. 11, No. 2 (1997), pp. 115-129.
- [8] J. Frey, T. Tannenbaum, M. Livny, I.T. Foster, and S. Tuecke, "Condor-G: A Computation Management Agent for Multi-Institutional Grids". *Cluster Computing*, Vol. 5, No. 3 (2002), pp. 237-246.
- [9] Hayek, F.A.v., W.W. Bartley, P.G. Klein, and B. Caldwell. *The collected works of F.A. Hayek*. University of Chicago Press, Chicago, 1989.
- [10] Huberman, B.A. *The Ecology of Computation*. North-Holland, Amsterdam, 1988.
- [11] IBM Corp., *Autonomic Computing*. IBM, Yorktown Heights, NY 2001. <http://www.research.ibm.com/autonomic/>
- [12] JavaSim Project. "JavaSim". *Ohio State University EEng Dept*. <http://www.javasim.org/>.
- [13] Pruitt, D.G. *Negotiation behavior*. Academic Press, New York, 1981.
- [14] Ripeanu, M., Peer-to-Peer Architecture Case Study: Gnutella Network. *University of Chicago*, Chicago 2001. <http://www.cs.uchicago.edu/~matei/PAPERS/gnutella-rc.pdf>
- [15] Sandholm, T.W., *Negotiation Among Self-Interested Computationally Limited Agents*. Ph.D. Thesis, University of Massachusetts, Amherst, 1996.
- [16] Smith, A. *An inquiry into the nature and causes of the wealth of nations*. Printed for W. Strahan; and T. Cadell, London, 1776.
- [17] Smith, R.E. and N. Taylor, "A Framework for Evolutionary Computation in Agent-Based Systems". *Proceedings of the 1998 International Conference on Intelligent Systems*. ISCA Press, 1998. <http://www.ics.uwe.ac.uk/~rsmith/fecabs.pdf>.
- [18] Tesfatsion, L., "How economists can get alive". Arthur, W.B., Durlauf, S., and Lane, D.A. (eds.). *The Economy as a Evolving Complex System II*, Addison Wesley, Redwood City, CA 1997, pp. 533-564.
- [19] J. Waldo, "The Jini Architecture for Network-centric Computing.". *Communications of the ACM*, Vol. 42, No. 7 (1999), pp. 76-82.