February 2007

Evaluation of Economic Resource Allocation in Application Layer Networks - A Metrics Framework

Werner Streitberger
University of Bayreuth, streitberger@uni-bayreuth.de

Torsten Eymann
University of Bayreuth, eymann@uni-bayreuth.de

Daniel Veit
University of Mannheim, veit@uni-mannheim.de

Michele Catalano
University of Ancona, catalano@dea.unian.it

Gianfranco Giulioni
University of Chieti-Pescara, guilioni@dea.unian.it

See next page for additional authors

Authors
Werner Streitberger, Torsten Eymann, Daniel Veit, Michele Catalano, Gianfranco Giulioni, Liviu Joita, and Omer F. Rana

In: Oberweis, Andreas, u.a. (Hg.) 2007. *eOrganisation: Service-, Prozess-, Market-Engineering*; 8. Internationale Tagung Wirtschaftsinformatik 2007. Karlsruhe: Universitätsverlag Karlsruhe

ISBN: 978-3-86644-094-4 (Band 1)
ISBN: 978-3-86644-095-1 (Band 2)
ISBN: 978-3-86644-093-7 (set)

Evaluation of Economic Resource Allocation in Application Layer Networks – A Metrics Framework

Werner Streitberger, Torsten Eymann
Chair of Information Systems
University of Bayreuth, Germany
95440 Bayreuth
{streitberger,eymann}@uni-bayreuth.de

Daniel Veit
Chair of Information Systems, E-Business and E-Government
University of Mannheim, Germany
veit@uni-mannheim.de

Michele Catalano
Department for Economics
University of Ancona, Italy
catalano@ dea.unian.it

Gianfranco Giulioni
Department of Quantitative Methods and Economic Theory
University of Chieti-Pescara, Italy
guilioni@ dea.unian.it

Liviu Joita, Omer F. Rana
School of Computer Science and the Welsh eScience Centre
University of Cardiff, UK
{l.joita,o.f.rana}@ cs.cardiff.ac.uk

Abstract

Support for economic resource allocation in Application Layer Networks (such as Grids) is critical to allow applications and users to effectively exploit computational and data infrastructure as a utility. The evaluation of resource allocation strategies plays a major part in the selection of a resource allocation method. This paper presents an evaluation framework for resource allocation methods in Application Layer Networks that attempts to support both a technical and an economic evaluation. The model uses a layered metrics pyramid with different aggregation levels and statistical methods.

On top of the pyramid, only one number, the social utility, is able to characterize an economic resource allocation method. This single number may serve to compare different resource allocation strategies. However, one can also determine values obtained for metrics at intermediate levels. We demonstrate using a prototype application how such a metrics pyramid may be deployed in reality, focusing on the implementation of the lower levels of the pyramid.

1 Introduction

An Application Layer Network (ALN) describes an abstraction that integrates different overlay network approaches, like Grid and Peer-2-Peer systems on top of the physical connectivity provided by the Internet. A common characteristic of ALNs is the redundant, distributed provisioning and access to data, computation or application services, while hiding the heterogeneity of the service and resource network from the user. One of the main issues in the development of future ALNs is the efficient exploitation of services and resources available in the network.

A key requirement of ALNs is to support scalable, dynamic, and adaptable allocation mechanisms. This issue is being addressed by a number of Grid and P2P projects such as Globus [FoKe97], Legion [CKKG99], Nimrod/G [BuAG00], CATNETS [EACC05], and Gnutella [AdHu00]. Although considerable progress has been made developing software architectures and allocation methods, which allow clients to obtain services "on demand", the evaluation of these methods with respect to each other is rarely undertaken. What is missing is a standard set of metrics that could be used as the basis for such an evaluation. It is important that such metrics allow comparison of both the application "behaviour", along with the infrastructure behaviour.

Currently, no metric framework exists that takes into account the characteristics of applications that may be deployed over ALNs – especially ones that measure the performance of the resource allocation strategy being used. Current Grid and Peer-2-Peer applications often use a service-oriented architecture, which is characterized by dynamic and heterogeneous resources. In such environments one of the key issues is the assignment of resources to services. This paper assumes that resource allocation is being undertaken in a "market" based environment – allowing service users and providers to sell and buy services and resources [ReSE06]. It is also assumed that two concurrent markets co-exist – a "resource market" and a "service market". A service instance can be provisioned by different possible, even heterogeneous resources; this heterogeneity, however, is shielded from the user, who acquires services on a virtualized, homogeneous service market. The presence of a market allows to use economic concepts with reference to the allocation strategy being adopted [ENRS06], e.g. different auction types, bilateral bargaining, market-oriented programming or computable genereal equilibrium approaches.

The characteristics of e.g. Grid and P2P applications in ALNs define particular resource allocation requirements which have the following characteristics [MoBa03]:

- Dynamicity: There are changing environments and there is a need for adaptation to changes.
- Diversity: Requests may have different priorities and responses should be assigned according to them. It is therefore possible for a number of different types of requests to co-exist, and determine how these should be handled becomes significant.
- Large scale: There are often a large number of nodes, and it is often necessary to group these elements to create hierarchical organisations.
- Partial knowledge: It is not possible to determine at any point in time the full state of the system or network.
- Complexity: Learning mechanisms are necessary to adapt to changes in infrastructure, and optimal solutions are not easily computable.
- Evolutionary: The system is open to changes which cannot be taken into account in the initial setup.

The goal of the evaluation framework presented here is to define a general set of metrics for performance analysis, which can be used in projects evaluating and comparing the performance of resource allocation mechanisms running in network exhibiting the above characteristics. Such a framework should include both technical parameters (such as quality of service factors – like response time) and economic parameters (such as overall cost of computation or data access), to compare different resource allocation methods with each other.

This paper is structured as follows: Section 2 presents the metrics used for the evaluation framework. Section 3 discusses how these metrics may be used for a technical and economical evaluation. A demonstration of the approach using a Grid prototype application is presented in Section 4, and a summary of the approach is provided in Section 5.

2 The Metrics Pyramid

It is often useful to be able to compare two allocation methods using a single index or number. Such an index provides an aggregated behaviour of an allocation method with reference to a number of features. The application of statistics is necessary to achieve the index, but is useful only in conjunction with a detailed measurement framework. The results of the measurement are intended to allow providing feedback on system behaviour, improve chances of successful adaptation, improve implementation, and increase accountability.
Figure 1 shows the logical structure of data and indices. It sums up the underlying methodology establishing the shape of a pyramid. Data are the basic units of information, collected through technical monitoring of the application layer network. Parameters which are likely to be of significance within the application and the resource allocation mechanism have to be selected for measurement. These parameters define the raw disaggregated data. To ease the analysis of the raw data, they are collected from different experiments (simulation runs) into a database. Disaggregated indicators provide the first stage of evaluation, and may comprise of a number of independently measured values. They help to improve the implementation of the resource allocation mechanisms.
For further evaluation, it is obligatory to take into account a set of characteristics that are not directly comparable because these characteristics correspond to variables of different dimensions and unit of measurement. Thus, they have to be made comparable, e.g. by

normalization, and then grouped into indicators. An indicator is defined as a ratio (a value on a scale of measurement) derived from a series of observed facts, which can reveal relative changes as a function of time. They provide feedback on system behaviour and also allow the analysis of performance and predictions of future performance.

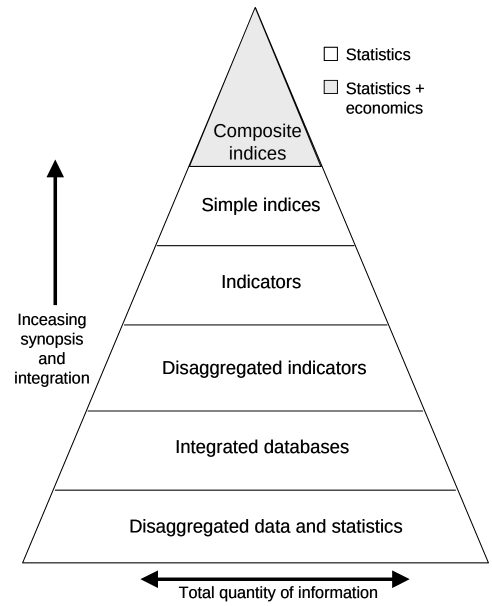

Figure 1: General method to obtain a composite index [PiZC00]

Finally, the simple and composite indices are computed, which represent performance benchmarks. They express information in ways that are directly relevant to the decision-making process. In general, indicators help the assessment, the evaluation, and most important, they help to improve accountability. Examples of applications using these techniques can be found in social sciences and in economics [Horn93][CoHR95][UNDP99].

Although highly aggregated indices are attractive because of their simplicity, they also carry risks. Most of all, aggregates tend to mask real-life complexity and detail for policy-making. Highly aggregated indices are important in giving a view of overall progress, but they should be readily disaggregated into its components that may help to specify reasons for the index going up or down and also answer questions of interest to decision-makers working on lower scales. The pyramid with its different layers supports this analysis progress, it enables both analysis at lower scales and decision-making on highly aggregated indices.

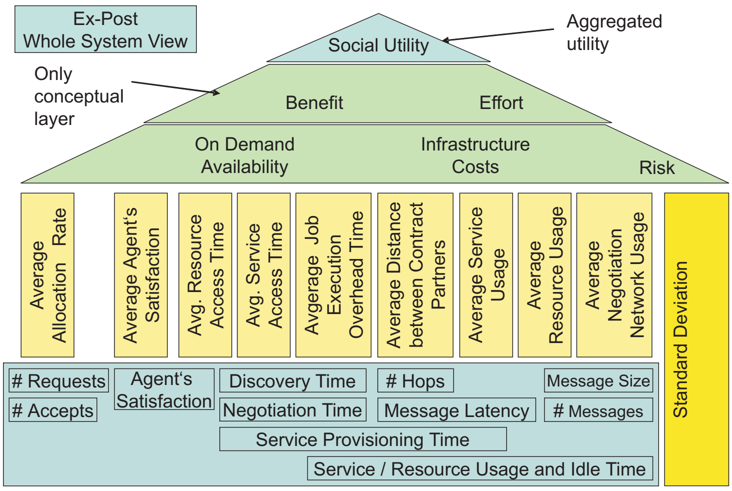

Figure 2: Metrics Pyramid

To support both technical and economic parameters, the evaluation process is divided into two layers. The first uses basic statistical concepts, while the second uses economic principles. The technical parameters, at the lower end of this pyramid, provide the basis for economic parameters that lie higher up in the pyramid, as illustrated in figure 2. A prerequisite is, that the key parameters at the bottom layer can be technically measured in the system to be analysed.

Aggregate indices may be generated out of several of these low level parameters. The parameters at the higher layers of the pyramid have economic semantics and therefore a higher significance. At the top of the pyramid there is a composite index defined as (global) “social utility” – providing a single metric to specify the overall performance of the allocation strategy being used in a particular application.

The selection of the technical parameters, and their aggregation to simple and composite indicators, can vary, depending on the resource allocation method and/or the measurable technical metrics of the application domain. The framework gives the possibility to add and remove metrics to optimize the framework for a special application domain without changing the general concepts used to build up this pyramid.

3 Technical and economical metrics

This section defines and illustrates various layers and aggregation steps of the metrics pyramid, (cf. figure 2), as used in the project CATNETS [EACC05] to analyse centralized and decentralized economic resource allocation methods.

The approach to technical metrics focuses on providing generic, easy to measure parameters, which can subsequently be aggregated. Technical layer metrics can be classified into: (I) efficiency measures (e.g. number of requests, number of acceptances); (II) utility measures (e.g. agent satisfaction); (III) time metrics (e.g. discovery time, negotiation time, service provisioning time) which are measures of the rate of change of market processes; and message-based metrics (IV) to measure the activity of users to communicate to find resources and services. The technical metrics include:

- Number of Requests: This metric measures demand, counted as the number of requests for services and resources.
- Number of Acceptance: The number of acceptance measures successfully requested (acknowledged) services and resources.
- Agent Satisfaction: The individual agent satisfaction measures the utility gain of a single transaction, which is the distance between his lowest price willing to pay in this transaction and the final price of the agreement. It is defined as the ratio between the subjective reservation value and transaction price.
- Discovery Time: This metric is used to measure the time to find a set of possible negotiation partners.
- Negotiation Time: The negotiation time measures the time needed to finish the negotiation between one buyer and one or several sellers. The measurement of the negotiation time starts after service discovery has completed, and ends before service provisioning.
- Service Provisioning Time (Effective Job Execution Time): The evaluation framework defines the service provisioning time as the service usage time of one transaction. It optionally includes also setup times, etc.
- Hops: The number of hops describes the distance between a service consumer and a service provider, counting the number of hops a message needs to traverse between them (this may be averaged over all the messages exchanged between a consumer and a provider).
- Message Latency: The message latency measures the time a message needs to arrive at the communication partner. Message latency is a parameter that indicates the performance of the physical network link and its message routing capabilities. It is expected that a large distance (cf. also number of hops) between a service consumer and

service provider should lead to higher message latency. Whether hops or latency is used to measure distance should be decided casually.

- Message Size: This metric is used to measure the message size (e.g. in kilobytes).
- Number of Messages: This value counts the number of messages needed for service allocation. The traffic on a physical network link could be computed by multiplying the message size with the number of messages sent.

These metrics are then used in two aggregation steps, as described below.

3.1 Simple indicators first aggregation

This layer defines a set of independent metrics which are normalized between 0.0 and 1.0. This makes it easier to find valid functions for the layers above, such as on demand availability and infrastructure cost. The technical metrics may be combined to obtain a framework that enables evaluation of different service oriented architectures. The aggregated metrics at the simple indicator layer are:

- Allocation Rate: This metric is a measure of the efficiency of the allocation process, which is computed using the number of requests and number of accepts. A buyer can demand services, but there is no guarantee that the allocation mechanism (centralized or decentralized exchange) performs a match between demand and supply.

$$allocation\_rate = accepts/requests$$

- Agent Satisfaction: This metric implicitly shows the average surplus of the service provider or user (agent) in the system (normalized to the interval 0.0 and 1.0). A low value means that an agent has not been able to complete its goals successfully during the negotiation process. A high value means that the agent can constitute good results satisfying its requirements.
- Access Time: This indicator evaluates the time needed from the starting point of discovery until the final delivery of the service.

$$access\_time = discovery\_time + negotiation\_time + provisioning\_time$$

- Job Execution Overhead: This is the additional time needed for negotiation. It refers to the overhead introduced during the service negotiation process. The overhead is the sum of the service access time and the resource access time.

$$job\_execution\_overhead = access\_time\_service + access\_time\_resource$$

- Distance between Contract Partners: Message latency is the messaging time incurred by agents, and it is proportional to the distance between the sending and receiving nodes. The latter would be the ratio between the actual distance and the maximum distance between agents.
- Service/Resource Usage: The network usage will be evaluated by the ratio between provisioning time and the total simulation time. This evaluation would be conducted for both services and resources.

$$service\_usage = provisioning\_time / simulation\_time$$

$$resource\_usage = provisioning\_time / simulation\_time$$

- Network Usage: This metric is used to measure the total number of messages exchanged between two agents.

3.2 Composite indicators - second aggregation

Composite indicators are an aggregation of simple indicators, and are closer in scope to the application than to the infrastructure. Simple indicators are normalized between a [0; 1] interval in order to be comparable with each other. From the economics perspective, composite indicators measure the benefit from exchange (on-demand availability), and costs incurred from activities in the market (infrastructure costs). An optimal allocation mechanism would have an on-demand availability value = 1, and infrastructure costs = 0.

On DeMand availability (ODM) is a composite indicator obtained as mean of simple indicators, and may be derived as:

$$ODM = 1/3\ (allocation\_rate + agent\_satisfaction + job\_execution\_overhead)$$

Infrastructure Cost (IC) is calculated in the same way. It is also a mean of multiple simple indicators, and may be derived as:

$$IC = 1/4\ (distance + service\_usage + resource\_usage + network\_usage)$$

It may be possible to model some of these metrics as stochastic variables, giving a mean and standard deviation over which the given metric varies. In economic applications, variance would be a measure for the overall “risk” to achieve stability of a given metric development.

3.3 The social utility - a composite index

To be able to use economic concepts into the evaluation process, a virtual policy maker is introduced, who aims to optimize the allocation. The behaviour of the policy maker is described with a goal function which is to be minimized under some constraints. The constraints are determined by the structure of the economy, i.e. the underlying laws of the economy. The policy maker does not have complete knowledge of the economy, but uses a model of the economy based on statistical cause-and-effect relations. The policy maker collects all the metrics from the previous layer, has some preferences about ODM and IC and is aware about the distribution of profits and costs. Analogies exist in macroeconomic theory, and are used here with regard to two related models: Barro Gordon [BaGo83] and Poole [Pool70].

Mapping these models to the evaluation framework leads to the goal of improving the utility of society and simultaneously lowering the fluctuation in the economy. If the policy maker minimizes the inverse ODM and IC by reducing the distance to zero of those variables (optimal values), the social utility is improved. Moreover, he knows that inverse ODM and IC are distributed across the population (i.e. he knows the statistical moments of those variables) in a sense that he looks at the distribution of benefits and costs over the agents. A result of the final optimization rule is that the policy maker is interested in minimizing inequality (measured with respect to the metrics defined previously) between agents.

In order to follow this approach for ALNs, a loss function is built using the composite indicators ODM and IC. ODM and IC are taken as stochastic variables and their distribution across a population (the set of service users and providers) by the first and second moments are considered. Renaming the variables for the presentation X (inverse on demand availability index) and Y (costs index), the ALN loss function is

$$L = E[\alpha(X - X^*)^2 + \beta(Y - Y^*)^2],$$

where α and β are weights and X^* and Y^* are target values. It is important to give a value to the targets. As X is the inverse of the On Demand Availability it can be concluded that a lower value of X is better for the policy maker. Thus, the target value is set equal to zero (X^*= 0). Y is a direct measure of costs so it is natural to choose zero also for this variable (Y^* = 0). With this target, the loss function is

$$L = E[\alpha X^2 + \beta Y^2]$$

The random variables can be expressed as

$$X = \mu_X + u \ \ and \ \ Y = \mu_Y + z,$$

where u and z are random variables with zero means and positive variances ($\mu_u = 0$, $\mu_z = 0$, $\sigma_u^2 > 0$, $\sigma_z^2 > 0$). Note that variances σ_u^2 and σ_z^2 are the variance across a population. Subsequently, the final social utility index may be specified as:

$$L = \alpha\mu_X^2 + \alpha\sigma_u^2 + \beta\mu_Y^2 + \beta\sigma_z^2 .$$

So one possibility for evaluating ALNs is to make use of:

$$\min \alpha\mu_X^2 + \alpha\sigma_u^2 + \beta\mu_Y^2 + \beta\sigma_z^2 .$$

This means that for the final evaluation of the scenarios, the first and second moments of ODM and IC have to be evaluated, whereas the first moment is the mean of ODM and IC, and second moment the variance of ODM and IC. In fact a policy maker has preference to minimize the inverse of ODM, thereby leading to the minimization of costs and the minimization of variability, i.e. the fair distribution of social utility between agents.

Applying the depicted metrics pyramid, a layered evaluation of different resource allocation methods is possible. The Grid application prototype, which is illustrated in the next section, uses the pyramid for a first evaluation of different scenarios. The focus of the evaluation is to show the relevance of this pyramid when being applied to a real Grid application prototype.

4 The Application Prototype

4.1 Application prototype description

The proof-of-concept prototype has been developed and named Catallaxy Data Mining (Cat-DataMining) [JPBG04]. Figure 3 shows the prototype components and related software agents as buyers and sellers in the Grid service market and Grid resource market. The prototype is composed of four main components: the Master Grid Service (MGS) – a Complex Service requestor; the Catallactic Access Point (CAP) – the entry point to the Complex Services for the Master Grid Service; the Data Mining Services – as types of Complex and Basic Services (BS); and job execution resources – as computational resources. The Complex Service (CS) agent acting on request of the MGS is the buyer entity in the service market, and the Basic Service is the seller entity in the service market.

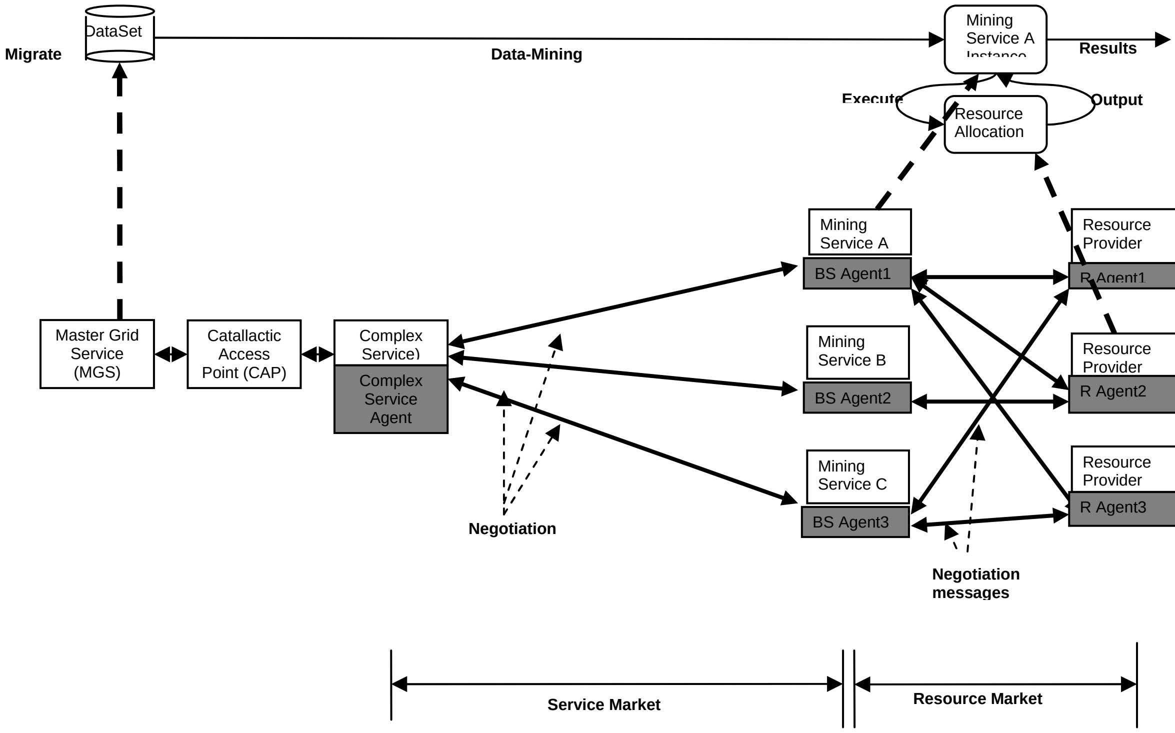

Figure 3: Cat-DataMining prototype – the Catallactic agents and the markets – one service invocation

The BS consists of a data mining job execution environment, which offers the deployment of multiple "slaves" able to execute the data mining task. Within the Cat-DataMining prototype, the data mining BS needs to be able to provide a response time as an important characteristic. With this goal the data mining BS agent buys resources in the resource market. Resource seller entities are able to provide a set of resources via the Local Resource Manager (LRM). The Resource Agents act on behalf of these Local Resource Managers, and provide an interface to the physical resources they manage.

The data mining BS agent is the buyer entity in the resource market, and the LRM agents are the seller entity on the resource market. The main functionalities of BS agents at the resource market are: (I) co-allocation of resources (resource bundles) by parallel negotiation with different resource providers (local resource manager entities); (II) informing the Complex Service about the outcome of resource negotiation.

4.2 Test-beds and preliminary results

We have performed several experiments in a cluster with the current version of the middleware, allowing for a preliminary performance assessment of the Grid Market Middleware (GMM) developed in the CATNETS project [JRCC06]. The goal of the experiments is to evaluate the autonomic behaviour of the GMM in terms of self-organisation, given by decentralized resource discovery, and in terms of self-optimisation, given by adaptation to load and capacity of the resources.

For these experiments we have used for the economic agents an implementation of the ZIP (Zero Intelligence Plus) agents [PrTo98] which use a gradient algorithm to set the price for resources. Clients initiate negotiations with a price lower than the available budget. If they are not able to buy at that price, they will increase their bids until either they win or reach the budget limit.

Services will start selling the resources at a price which is solely influenced by the node's utilization, following the pricing model presented in [YeBu04]. As services get involved in negotiations, the price will also be influenced by demand. If a service agent is selling its resources, it will increase the price, when resources are sold to identify the price the market is willing to pay. When it no longer sells, it will lower the price until it becomes competitive again or it reaches a minimum price defined by the current utilization of the resource.

In order to test the performance of the market based resource allocation mechanism, we setup controlled experiments deploying several instances of the middleware in a Linux server farm. Each node has 2 CPU Intel PIII 1 GHz and 512 MB of memory. The nodes in the farm are connected by an internal Ethernet network at 100Mbs.

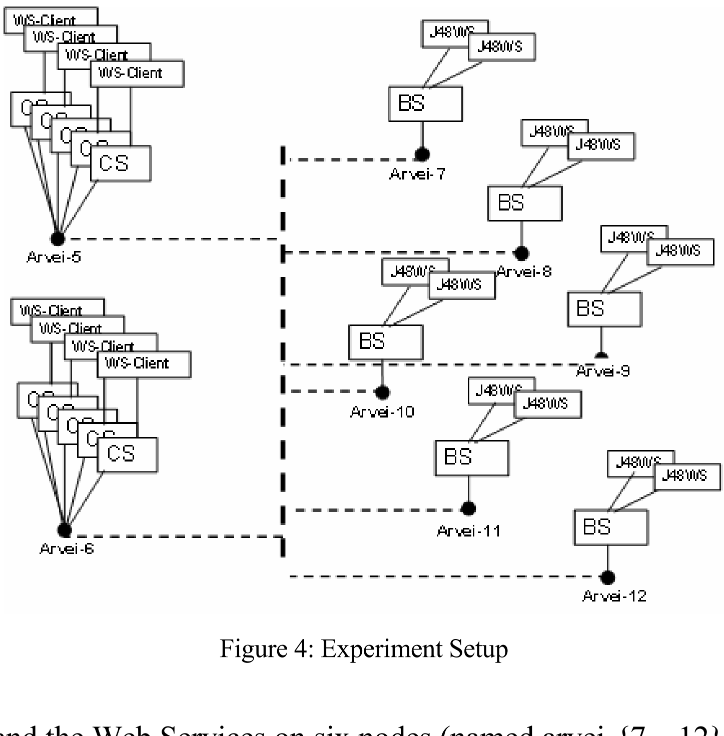

Figure 4: Experiment Setup

We deploy GMM and the Web Services on six nodes (named arvei-{7…12}). On each node we also deployed a Web Service which performs a CPU intensive calculation on the machines, increasing load. These Web Services are exposed in a Tomcat server. Access to execute these

Web Services on the Resources is what will be negotiated by the Service with the Clients. Figure 4 shows a schematic view of this deployment.

The experiments consisted in launching 2 Clients concurrently from 2 other nodes which were not running the Web Services. We generate a baseline load on all three nodes of 25% of CPU usage to simulate some background activity. Each Client performs 50 requests, in intervals of 10 seconds. Whenever a Client wins a bid with a Service, it invokes the Web Service in the selected node. The complete experiment runs for about 10 minutes. To better test autonomic load balancing, we artificially stressed one of the nodes (arvei-10) up to 95-100% of CPU usage for a short time during the experiment.

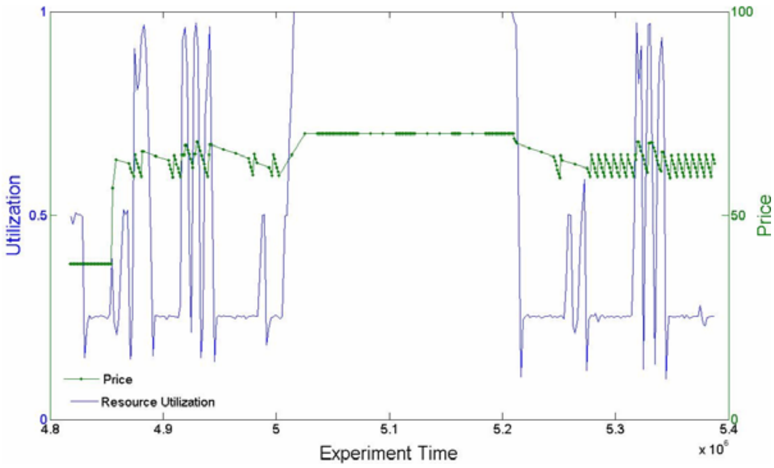

Figure 5: Experiment Results – Price and Utilisation Basic Service at node arvei12

Figure 5 and Figure 6 show the measured results of selected nodes used in the experiment. The measured metric price corresponds to the agent satisfaction of the presented metrics pyramid, the execution time to the provisioning time, and the utilisation metric to the resource usage. The metric negotiation time equals the negotiation time in the metrics pyramid. Both figures show the same experiment time interval. High utilisation of the CPU leads to higher prices of the resource CPU and to higher execution times, whereas the negotiation time doesn't significantly increase. Prices of the CPU are lowered time-delayed; a short increase of the utilisation increases the prices again. As expected, a lower utilisation of the CPU decreases the execution time.

Essentially, each data for each technical metric is referring to a single transaction and it has to be collected by the corresponding agent. Price and utilization data are collected periodically and independently by each transaction, giving a number of observations whereas execution time, negotiation time, and allocation are a dataset of 100 observations, which are collected by the complex services.

The computation works as follows: the first price data that has been collected after the execution data time stamp has been considered. Repeating the above procedure for each execution time, an observation data set of 100 items both for price and utilization has been extracted.

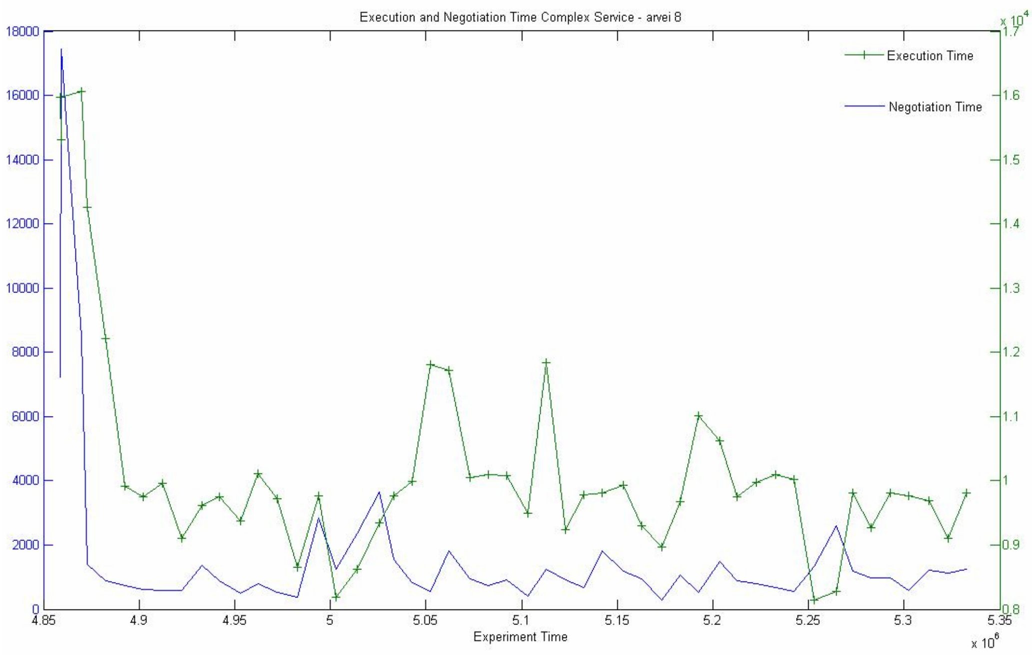

Figure 6: Experiment Results - Execution and Negotiation Time at node arvei8

The first aggregation level contemplates the construction of the normalized metrics set agent satisfaction, access time and resource usage. Agent satisfaction and resource usage are normalized to the interval between 0.0 and 1.0. The access time is normalized as follows:

$$access.time = e^{-\beta(execution.time + negotiation.time)}$$

where the weighting factor $\beta = 0.00005$ is chosen arbitrarily. The on demand availability is defined the sum of the benefit metrics access time and satisfaction and the infrastructure cost indicator equals the resource usage. Applying the mean and the standard deviation of both on demand availability and infrastructure costs (see Table 1), we obtain the final index like follows:

$$L = \alpha\left[1 - 1/2\left(\mu_{access.time} + \mu_{agent.sat}\right)\right]^2 + \alpha(1/2)^2\left(\sigma^2_{agent.sat} + \sigma^2_{access.time}\right) + \beta\left(\mu_{resource.usage}\right)^2 + \beta_{resource.usage}$$

Metric	*Value*	*Metric*	*Value*
L	*0,43508*	α β	*0,5*
$\sigma_{resource.usage}$	*0,32025*	$\mu_{resource.usage}$	*0,48686*
$\sigma_{agent.sat}$	*0,015615*	$\mu_{agent.sat}$	*0,46295*
$\sigma_{access.time}$	*0,013218*	$\mu_{access.time}$	*0,080522*

Table 1: Experiment Results - Final Index

5 Conclusion

The design of a general evaluation framework for resource allocation in ALNs has been presented, which consists of a set of technical and economic parameters. The parameters in the lower levels of the framework provide technical data, while higher level economic parameters could be integrated into business models for decision makers.

However, the implementation and possible refinements of the presented concept is ongoing work which is not yet finalized. An exact definition about how to measure the defined metrics for the framework and integrate them is also ongoing work. A first implementation of the presented metrics framework has been discussed – work undertaken as part of the EU CATNETS project. This project develops a resource allocation architecture based on decentralized economic models, allowing resource management polices to be specified in a decentralized, autonomous and infrastructure independent way. The economic concepts of the CATNETS project uses bargaining mechanisms improving the resource allocation in Application Layer Networks (ALNs). An application example is used to demonstrate how metrics associated with the evaluation framework can be specified.

Acknowledgements

This work has partially been funded by the EU in the IST programme “Future and Emerging Technologies” under grant FP6-003769 “CATNETS”.

References

[AdHu00] Adar, Eytan; Huberman, Bernardo A.: Free riding on gnutella. Technical report, Xerox PARC, 10 Aug. 2000.

[BaGo83] Barro, Robert; Gordon David: Rules, discretion and reputation in a model of monetary policy. Journal of Monetary Economics, Vol. 12, No.1, pp. 101–122, 1983.

[BuAG00] Buyya, R.; Abramson, J.; Giddy D.: Nimrod/G: An architecture for a resource management and scheduling system in a global computational grid. In Proceedings of the 4th International Conference on High Performance Computing in Asia-Pacific Region (HPC Asia 2000), 2000.

[CKKG99] Chapin, S. J.; Katramatos, D.; Karpovich, J.; Grimshaw, A.S.: The legion resource management system. In Dror G. Feitelson and Larry Rudolph, editor, Job Scheduling Strategies for Parallel Processing, pages 162–178. Springer Verlag, 1999.

[CoHR95] Cobb, C.; Halstead, T.; Rowe J.: The genuine progress indictor: Summary of data and methodology. San Francisco: Redefining Progress, 1995.

[EACC05] Eymann, T.; Ardaiz, O.; Catalano, M.; Chacin, P.; Chao, I.; Freitag, F.; Gallegati, M.; Giulioni, G.; Joita, L.; Navarro, L.; Neumann, D.; Rana, O. F.; Reinicke, M.; Schiaffino, R.; Schnizler, B.; Streitberger, W.; Veit, D.; Zini F.: Catallaxy-based grid markets. In Proceedings of the First International Workshop on Smart Grid Technologies (SGT05), 2005.

[ENRS06] Eymann, T.; Neumann, D.; Reinicke, M.; Schnizler, B.; Streitberger, W.; Veit D.: On the Design of a Two-Tiered Grid Market Structure. Business Applications of P2P and Grid Computing, MKWI, 2006.

[EyPS00] Eymann, T.; Padovan, B.; Schoder, D.: The catallaxy as a new paradigm for the design of information systems. In Proceedings of the 16th IFIP World Computer Congress, Conference on Intelligent Information Processing, 2000.

[FoKe97] I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. The International Journal of Supercomputer Applications and High Performance Computing, 11(2):115–128, 1997.

[Horn93] Horn, R.: Statistical Indicators For the Economic & Social Sciences. Cambridge University Press, 1993.

[JPBG04] Joita, L.; Pahwa, J.S.; Burnap, A.; Gray, P.; Rana, O.F.; Miles, J.: Supporting collaborative virtual organisations in the construction industry via the grid. In Proceedings of the UK e-Science All Hands Meeting 2004, Nottingham, UK, 2004.

[JRCC06] Joita, L.; Rana, O.; Chacin, P.; Chao, I.; Freitag, F.; Navarro, L. Ardaiz O.: Application Deployment using Catallactic Grid Middleware. In: Proceedings of the 3rd International Workshop on

Middleware for Grid Computing (MGC 2005), co-located with Middleware 2005, 2005, Grenoble, France.

[MoBa03] Montresor, A.; Baboglu, O.: Biology-inspired approaches to peer-to-peer computing in bison. In Proceedings of the 3rd International Conference on Intelligent System Design and Applications (ISDA'03), Advances in Soft Computing, pages 515–522. Springer-Verlag, Aug. 2003.

[PiZC00] Pinter, L.; Zahedi, K.; Cressman D.: Capacity Building for Integrated Environmental Assessment and Reporting. International Institute for Sustainable Development and United Nations Environment Programme, 2000.

[Pool70] Poole, W.: Optimal choice of monetary policy instruments in a simple stochastic macromodel. Quarterly Journal of Economics, 84:197–216, 1970.

[PrTo98] C. Preist, M. van Tol, Adaptive agents in a persistent shout double auction. In Proceedings of the First international Conference on information and Computation Economies (Charleston, South Carolina, United States, October 25 - 28, 1998). ICE '98. ACM Press, New York, NY, 11-18.

[ReSE06] Reinicke, M.; Streitberger, W.; Eymann, T.: Evaluation of Service Selection Procedures in Service Oriented Computing Networks. In: Multi Agents and Grid Systems, Vol. 2, pp. 271-285, 2006.

[UNDP99] UNDP. Human development report. New York: Oxford University Press, 1999.

[YeBu04] C.S. Yeo, R. Buyya, Pricing for utility-driven resource management and allocation in clusters. Proceedings of the 12th International Conference on Advanced Computing and Communication (ADCOM 2004), Ahmedabad, India, December 2004. Allied Publisher: New Delhi, India; 32–41.