Trading computing power with ReGTime

Bernd Dreier [a], Annja Huber [a], Theo Ungerer [b] and Markus Zahn [a]

[a] *Institute of Informatics, University of Augsburg, D-86 135 Augsburg, Germany*

[b] *Institute of Computer Design and Fault Tolerance, University of Karlsruhe, D-76 128 Karlsruhe, Germany*

Abstract

ReGTime [1] (*Re*nt *G*igaflops some*Time*s) is a software package for an envisioned "computing power market". ReGTime helps customers to search for providers who allow their workstation clusters to be rented. It manages the rental of disposable machines on provider's side and establishes the contract with the customer. It is responsible for granting access to rented systems and observes compliance with established contracts.The organization of such a computing power market is derived from trading mechanisms of existing electronic markets.

Keywords: cluster computing, electronic market, distributed computing, ReGTime

1 Introduction and Motivation

Distributed computing platforms consisting of servers, workstations and other computers connected by local and wide-area networks are in common use. The combined computing power of distributed platforms can be harnessed for a broad range of applications by *distributed computing*, where a complex task is split in several smaller tasks executed in parallel. In general, parallel algorithms known from parallel computers can be executed using a distributed computing platform. In combination with high-speed networks, distributed computer systems offer the possibility to combine a team of computers to *one virtual machine*.

Due to the actual communication performance, distributed computing in wide area networks is not commonly applied yet, but active research is done under the buzzwords "Metacomputing" [1–3] or "Hypercomputing" [4]. However, the increasingly higher bandwidths of wide-area networks lay the foundation for the common usage of remote resources.

[1] ReGTime is available at http://www.informatik.uni-augsburg.de/info1/regtime/

Today's research in metacomputing is mainly directed towards the connection of two or more supercomputers — multiprocessors or vector computers — through a high-speed wide-area network. One example is the "Virtuelle Rechenzentrum Südwest" [5] coupling the IBM SP2 of the University of Karlsruhe with a vector computer SNI VPP300, another one the connection of an Intel Paragon XP/S10 of KFA Jülich and an IBM SP2 over an ATM-connection [6]. Even larger collections of supercomputers were assembled by the IWAY experiment (Information Wide Area Year) and Globus Project [3] in the United States and by the Legion project of a worldwide virtual computer that distributes computation creating the illusion for the users of a very powerful desktop computer [7]. The project "hypercomputing with workstation clusters" of the University of Rostock aims at the organization of a hypercomputer consisting of more than a thousand workstations of different institutions geographically distributed over Germany [4]. Applications are mainly scientific research projects, e.g. the n-body problem of non-destructive evolution processes inside a cluster of galaxies [8] or the 2D electromagnetic finite element mesh in physics [7].

Since distributed computer systems do not share a common address space, message-passing has become the programming paradigm of choice for parallel applications, especially in scientific domains. Message-passing programming is wide-spread in universities since several years as a method to program parallel computers and distributed computing platforms. The parallel programming environment PVM [9] (Parallel Virtual Machine), which is based on the message-passing model, allows portable distributed programming of workstation clusters and parallel computers. The emerging message-passing standard is MPI [10] (Message Passing Interface) which is up to now mainly directed towards multiprocessors. Until now, using networked computer systems as a virtual parallel computer is mostly found at universities and research institutions.

Selling computing power is already widely used in the areas of mainframes and today's parallel computers. In these areas there are only a few providers which are well-known to their customers. Access is provided via ftp, telnet or similar means. The (parallel) program is transferred to the remote computer, it is executed on the provider's machines and after termination the results are sent back to the customer. No distributed computing occurs.

The idea behind ReGTime is that managers of networked computers offer their free capacity on a "computing power market", and customers are able to rent resources on a short-term basis. We oriented ourselves on existing solutions and decided that the internet combined with the World Wide Web (WWW) is the appropriate basis for such a computing power market. The concept of ReGTime is derived from trading mechanisms [11–13] in electronic markets. Trading in electronic markets may be caused by various reasons e.g. prospective competition advantages, the expectation of obtaining profits and organizational needs [14]. Companies are able to temporarily access high com-

puting capacities without a high investment for a powerful computer system. The providers increase the profitability of their machines through an additional income.

2 Electronic Market for Computing Power

2.1 Electronic Markets

The enormous growth in the number of Internet users, the number of hosts already connected to the WWW, and the number of companies establishing a Web presence influence the development of new markets. As a result new products and services come into existence. With the help of modern information and communication systems, new applications are being built, e.g. electronic markets which support trading of goods or rendering of services. Electronic commerce encompasses all forms of electronic interaction between various businesses. The National Information Infrastructure (NII) performs the following functions [15]:

- bring products into the market (e.g. Research & Development via telecommunications),
- match buyers to sellers (e.g. electronic malls, Electronic Funds Transfer), and
- deliver electronic goods (e.g. information).

Networks can facilitate the coordination between buyers and sellers, reducing transaction costs, and therefore lowering the costs to the final consumers. Because of the anonymity of the global market, Sarkar [16] argues that transactions without any intermediation will be rare on an NII. Some examples demonstrate that the idea of an internet-based brokering service showed up in several contexts recently: FAST [17] is a project intending to manage the procurement of electronic parts. The BargainFinder [18] searches at several internet-CD-shops for cheap CDs. The MeDoc-project [19] supports the search of computer science related publications in the internet. The "Information Broker" of MeDoc takes requests of all kinds of information and answers with references where the information can be found. However, to our knowledge, none of today's brokering services is concerned with renting computing power, except for ReGTime.

2.2 Computing Power Market

The participants in computing power markets can be divided into three groups: providers, customers and brokers.

Providers are companies and organizations which own one ore more interconnected workstation clusters with occasionally unused computing power.

Especially suitable for rental are nighttimes and weekends. Potential candidates for rental are therefore all providers of workstation clusters with high idle times. One of the reasons for renting computing power in a market is additional income. Irrespective of whether the capacities are used or unused, they will cause fixed costs, e.g. power consumption, maintenance and depreciations. These costs may be reduced and profits may be gained. A price model can take into account the user group, the frequency of rental, the number of rented workstations, or the time of the conclusion of the contract.

One of the demands the application programs of customers have to meet are their distributed and well scalable structure. These programs should be parallel programs with message-passing based on a wide spread programming environment (like PVM or MPI). Further appliable environments are client/server-systems, e.g. DCE [20]. The technical prerequisite — a connection to the internet — is obvious. We imagine applications for the customer which need high computing power occasionally. Potential application programs are numerical simulation models. The availability of high computing power allows the customer to apply qualitative and quantitative better models, for instance by use of finite-element simulation models instead of simpler models, and by scaling these models far beyond the available computing power of a single workstation. Hence with renting computing power, strategical competition advantages could be achieved.

Besides the primary aims of additional income and competition advantages, providers resp. customers anticipate some further synergy effects. If the rental takes place within a virtual company [21] the business relations could be better organized with respect to speed-up of business transactions.

Proceeding from an existing computing power market including the anonymity we need an instance which brings together supply and demand of provider and customer. Such a brokering service handles information and communicates between these two participants. But to mediate offers is not enough, there is a need for a functionality that establishes contracts. If a contract will take place between a customer and a provider in our computing power market, the access to the computers has to be organized and the compliance to the contract has to be observed.

2.3 Authentication and Security in a Computing Power Market

The two criteria *authentication* and *security* are of prime importance in a computing power market. Protecting the internal data of a company is crucial for the participation in a computing power market. The security problem is generally known and we will show which measures could be taken for a solution.

To clarify the subject of authentication, we consider a well known example: To open a bank account, a customer must personally appear at the banking

house. Then, he has to prove his identity (i.e. show his identity card), that's what we call authentication. Finally, the customer signs a contract and as a result, from now on, the given signature will be used to check the customer's identity. In an electronic market, the signature is substituted by an electronic signature.

Considering the computing power market, different user groups require different handling of authentication which leads to adapted authentication mechanisms. It is self-evident to trust persons which are personally known, e.g. colleagues or business partners, meaning that authentication at a low security level (e.g. transmission of the electronic signature by phone or email) is sufficient. A more secure level of authentication is needed for customers, which are recommended from business partners but not personally known by the provider (e.g. submission of the electronic signature and a copy of the identity card by fax). The most secure level of authentication (e.g. video conference or personal appearance) should obviously be achieved for customers, which are outside users and e.g. mediated by an electronical broker only.

Very close to authentication is the security aspect of prying and manipulating data. Considering the insufficient security properties of today's computer systems granting access to an unknown customer is risky. This problem concerns both providers and customers. Through ReGTime, customers get access to conventional UNIX accounts. Using this account, a customer might be able to pry into the provider's data. One important precaution would be a machine administration that protects the computers with important data and programs through a firewall from the machines to let through ReGTime. The customer has the ability to protect his data *actively*, whilst the provider has to trust in the configuration of his general installation. In general, only parts of the end result of a distributed program are visible on the provider's machines. Therefore, prying seems not to be very profitable for providers. Additional protection is offered by cryptographic techniques or the use of secure, network-transparent filesystems like e.g. DCE/DFS. So we regard the aspect of security mainly as a problem of the provider. We assume, as in the field of authentication, that personally known customers do no harm, the more so as they are liable for damage [22]. It is much more risky to lease computers to outsiders. A solid authentication mechanism increases the hindrance for prying and sabotaging and allows liability of harming.

3 ReGTime

With ReGTime we introduce a new service in a newly created electronic market. With the help of ReGTime, customers needing high computing power are able to find providers allowing the rental of their under-utilized computers. The providers will grant the necessary rights based upon a contract, allowing

the customers to spread their distributed applications over the rented virtual machine.

ReGTime is independent of any distributed programming environment (e.g. PVM or MPI) the customers want to use. ReGTime could even broker non-distributed software like usage of hardware synthesis tools or numerical packages.

3.1 Concept of ReGTime

In the concept of ReGTime, there are three participants in the computing power market: provider, customer and broker. To be part of the market, the provider manages and offers his workstation clusters with ReGTime. The customer contacts a broker (in resemblance to existing market mechanisms) and asks for offers. The broker holds the information about providers willing to let their machines, thus, the broker acts as a mediator between customer and provider. The customer selects one of the offers, given as a reply from the broker. If the customer accepts an offer he has to conclude a contract with the provider. The broker himself is no partaker in that contract. Thereafter the customer is able to use the rented workstation cluster for distributed computing.

Institutions with one or several workstation-clusters are well suited to be providers. Corporations with computers linked via corporate networks are another target group — thereby improving the exploitation of their own resources.

Considering the fast progress of networking via internet and intranet, the prerequisites to offer and sell unused computing power on a market are fulfilled. This lays the foundations of the envisioned "computing power market". In such a market, many potential participants (providers and customers of computing power) exist. There is substantial demand for brokering, since the potential users of distributed systems still miss a service that provides easy access to rentable computing power. These are our reasons for creating ReGTime, a software package to offer, broker and rent computing power.

3.2 Architecture of ReGTime

ReGTime is designed to help both: customers to find free capacities in the computing power market and providers to offer their unused computing power to the market. The provider has to offer one or several workstation clusters for rental. Each workstation of a cluster is characterized by its hardware and software features and the time frames at which the machine is available for leasing. The provider specifies the computing nodes and the accounts available to the customers. To administer these informations easily, ReGTime provides

an easy-to-use configuration tool (see Figure 1) to set parameters like offered nodes, prices, and leasing time frames.

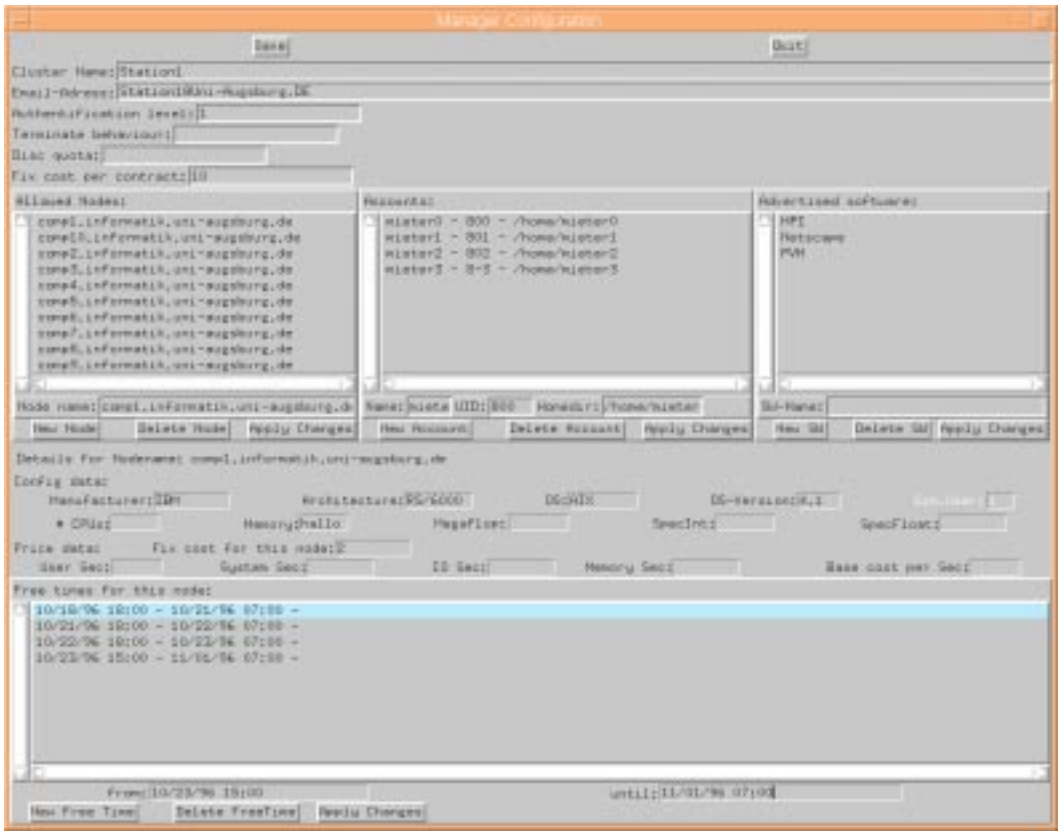

Figure 1. ReGTime's configuration tool for providers

After registering with some brokers, a provider is part of the computing power market. The registration is done automatically by the configuration tool, no further human interaction is necessary.

ReGTime's broker service uses an easy to use WWW interface to interact with the customers. Computational power is requested by specifying the requisites of his distributed application (e.g. time frame, number of nodes, performance requirements, price limits etc.) in a WWW form. Furthermore, the broker maintains a list of providers willing to let their computing power. Upon receipt of a customer's request, the broker queries all known providers and assembles the replies of the providers to one or several offers for the customer. The broker tries to minimize the number of different providers needed to satisfy a request.

Once a customer has accepted a specific offer, the broker establishes a contract between customer and the participating providers. After conclusion of the contract, the customer receives the necessary informations about the nodes and accounts he may use for his distributed applications in the given time frame.

In addition, providers can force their customers to sign the contract (see Figure 2, step 9), i.e. authenticate themselves with help of the public-key mech-

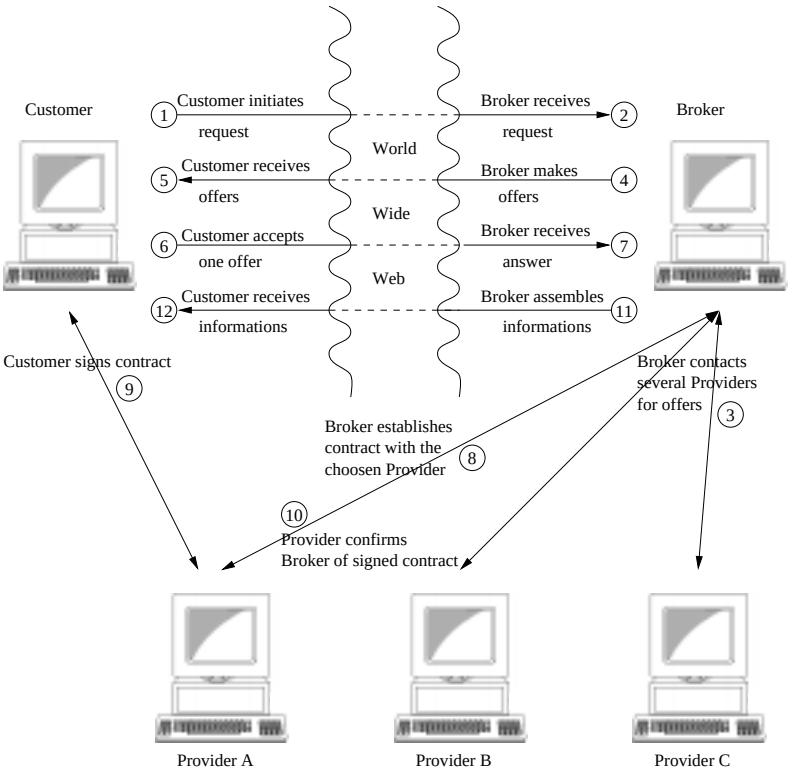

Figure 2. From request to contract with ReGTime

anism PGP (Pretty Good Privacy) [23]. Therefore, the provider is able to restrict the number of customers gaining access to his computer systems to a group of trustworthy people or organizations. The WWW request form contains a checkbox named PGP-authentication. If the customer is not willing to authenticate himself (i.e. does not select the checkbox) he will not receive any offers from providers who insist on PGP-authentication.

During a contract, the customer is authorized to access the rented machines, all customer activities are logged. Figure 3 illustrates the use of the newly gained computing capacities. These accounting informations are used to compute the invoice. ReGTime's accounting mechanism is based on the standard Unix accounting system, which measures CPU-time (kernel and user mode), average memory consumption and I/O-activities of the customer's processes. The prices for each category may be part of the contract. Alternatively, fixed costs for a contract and costs based solely upon the duration of a contract are also possible. After the period of lease, an invoice is sent to the customer and the account is closed.

3.3 A Session with ReGTime

To give an impression of ReGTime, we present a short guided tour of a session with ReGTime from request to invoice.

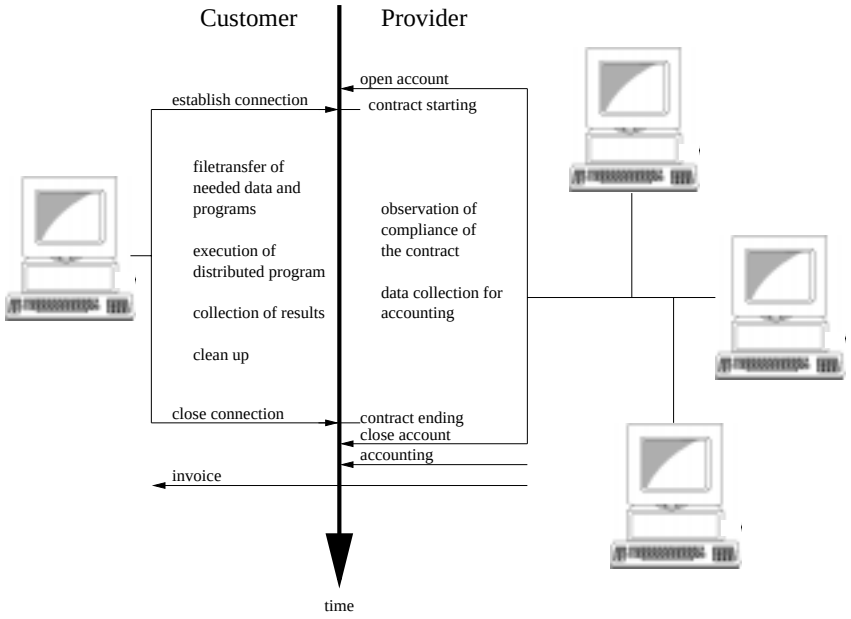

Figure 3. Activities of ReGTime during contract time

A customer requests computational power by entering parameters like number and types of acceptable computers, operating system, software prerequisites and the contract's time frame in a WWW form (see Figure 4). The time frame can be specified flexibly by an earliest and latest starting point and a duration. Additionally, price limits can be set for fixed costs, time-dependent costs, CPU-, memory-, and I/O-related costs individually.

As a first result, ReGTime's broker service returns a list of several offers. Some of these offers may combine several providers (see Figure 5). Combined offers are necessary if a request cannot be satisfied by a single provider. Whether the customer accepts a combined offer or not will strongly depend on the granularity i.e. the communication needs of the application which is to be distributed. If the customer selects an offer, a contract between him and the providers involved with this offer is established. If the provider insists on PGP-authentication and the customer has selected the appropriate checkbox, he will be prompted for his PGP-password to sign the contract with his private key.

At the starting time of the contract, access to the rented systems is provided by regular accounts. This allows e.g. PVM or MPI users to add these hosts to their parallel virtual machine for further distributed processing. Shortly before termination of the contract the customer is advised to finish working soon and to transfer back results.

At the end of the period of lease, all customer's processes on the rented machines are killed, the account is locked, the home directory is deleted, and an invoice is generated upon the accounting data. The invoice is a bill sent by email.

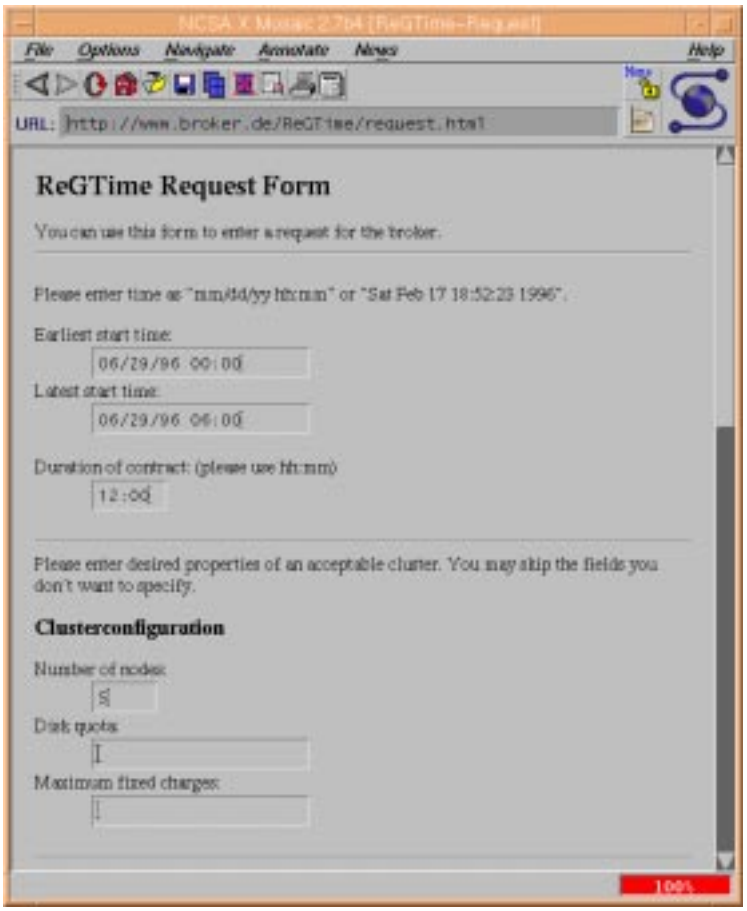

Figure 4. ReGTime's WWW request form

4 Classification of user models

ReGTime is designed to work in a worldwide market. We call the most common model of application the *outside company model*, because selling and renting computing power is done by companies unknown to each other. The only motivation for selling computing power is to gain profit. Hence, the costs of the selling company, which include the efforts for authentication and other security related tasks, must not exceed the potential profit. Due to the manifold security risks, these costs are very complicated to calculate. A consequent administration of the seller machines is crucial to reduce security risks. Whereas the seller could not expect advantages in competition or in his organization structure beyond his gain in profits the customer has competitive advantages due to his enlarged computing power.

In the following, we examine the use of ReGTime in restricted user models. The most important obstacle on the way to a broad acceptance of a computing power market are security risks which can't be completely excluded, if provider and customer are unknown to each other. Therefore, we restrict the circle of potential providers/customers.

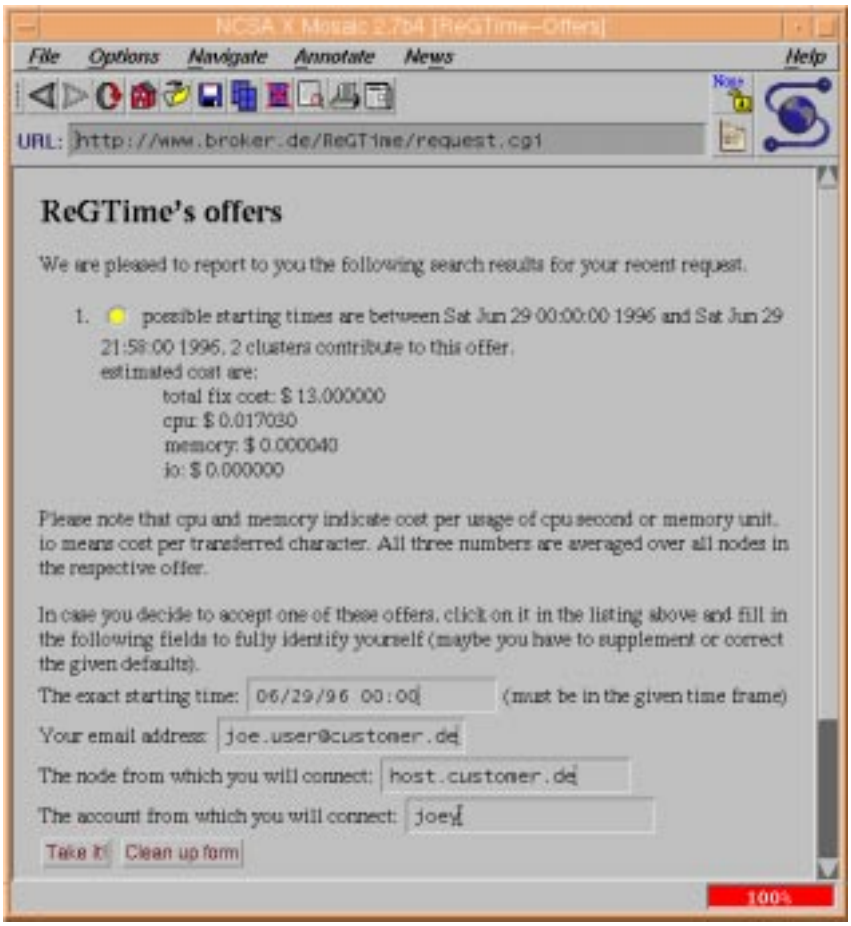

Figure 5. A sample offer of a contract

One restricted user model is to shift the previously global market into the boundaries of a *virtual company*. Provider and customer of computing power reside within this virtual company. A virtual company is based on a short-term union of singular companies with the purpose to exploit competitive advantages corporately [21]. The union of partners from universities or other institutes of research known to each other is another example for the virtual company model. Confidence within a virtual company should be consolidated by appropriate contracts. The partners in the virtual company harness previously unused computing power and become more competitive and profitable due to the increased quality of their products. Moreover, if partners in a virtual company work on complementary fields, the likelihood of partners spying out each other is reduced. The need for authentication, however, is not reduced, but it can take place through trusted communication channels established between the participating partners.

Further restricting the circle of possible providers and customers leads us to the model of *incorporate departments* — computing power is selled within a company. This model allows a further decrease of the security level, since all data and communication necessary for renting and using computing power

resides within the boundaries of a firewall [24], which protects the company against invaders from outside. Under-utilized resources can now be used to increase the product quality, e.g. by using more complex and efficient software. In big companies accounting between different departments of a company is done already. The accounting of rented computing power could be easily integrated in this scheme.

5 Implementation of ReGTime

ReGTime implements the tasks listed above by a set of three Unix dæmons (see Figure 6): *brokerd* acts as the broker i.e. it communicates with the customer by World Wide Web. The *providerd* is working as a provider's manager, it negotiates on behalf of a provider and establishes contracts. To observe compliance with the contract, a third dæmon acts on the provider's side, running on every rentable machine: *guardiand* mainly controls access, terminates processes at the end of a contract and computes invoices. For portability reasons most of the guardian's functionalities are implemented as Perl scripts [25].

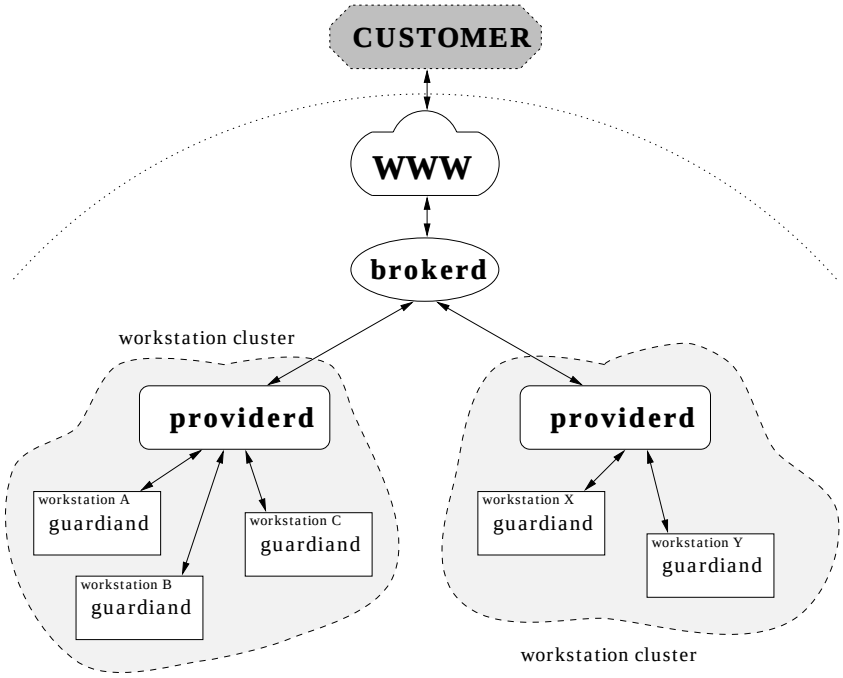

Figure 6. Implementation of ReGTime

ReGTime's customer interface is designed as a WWW form. Customers' input is forwarded (via sockets) to the associated broker (*brokerd*) by CGI scripts. Thus, the broker may run on a different host as the http-dæmon does. Every request is handled by a separate process forked by the broker.

Accounting information is gathered by the operating system of each rented machine. Today, almost all wide-spread UNIX systems offer a *pacct*-like accounting system. Using this system, the kernel generates information for every

terminating process, containing its run-time and resource consumption (like CPU-time, memory and I/O). As already mentioned, access to rented workstations is based on regular accounts, therefore the required informations about the customer's resource consumption can be derived from the standard accounting system. At the end of a contract, accounting information is gathered from the enlisted hosts and is used by a distinguished *guardiand* to assemble the invoice. The invoice is sent to the customer via email.

To provide a secure signature, a program called *sign-it* establishes an interface between ReGTime and PGP and the customer's site. If the customer selects PGP-authentication he will be prompted for his PGP-password to sign the contract with his private key. Furthermore, the contract is encrypted using the customers private key. On the providers side, PGP is asked to check the signature against a copy of the customer's public key. If this signature is validated, and both the original version of the contract and the encrypted one are identical, the provider considers the customer to be correctly authenticated and the contract is accepted.

Due to the method of PGP, it is essential under which prerequisites a provider is willing to add someone's public key to his key ring. Thus, a key received by a provider should be proofed using a second communication method. The highest security level could be achieved, if keys would only be accepted after a personal introduction with a valid identity card. A provider can choose any level of security somewhere along these lines. This additional effort is necessary only once to guarantee the identity of public keys. For informations about installation of ReGTime see [26].

6 Practical Results and Conclusions

ReGTime is a software package for an envisioned "computing power market". It consists of three main parts. Firstly, ReGTime helps customers to search for providers who lease workstation clusters. Secondly, it manages the leasing of disposable machines on provider's side and establishes the contract with the customer. Thirdly, it is responsible for granting access to rented systems and compliance with established contracts. Furthermore, it supports secure authentication between business partners by PGP signatures.

Testing of ReGTime on SUN SparcStations running Solaris 2.4 and on IBM RS/6000 workstations with operating system AIX 4.1 is successfully finished. Portability turned out to be good, because the implementations of ReGTime's components use widespread programming mechanisms (e.g. PERL and C) only. A first prototype of our implementation was presented at CeBIT'96 in Hannover. The demonstration system consisted of two independent workstation clusters at the universities of Augsburg (up to 20 IBM workstations) and Karlsruhe (four SparcStation5). The described clusters were accessed from the exhibition, an additional IBM RS/6000 on the fair ground acted as a third

cluster. ReGTime is easy to use, access to computing power is provided fast and comfortably. We have successfully tried several existing distributed applications (based upon PVM) in conjunction with ReGTime.

ReGTime is originally aimed at a worldwide open computing power market. However, we experience a certain retention of potential providers due to security objections despite the authentication mechanisms. We therefore envision an application of ReGTime coming up first among partners that are not totally unknown to each other — like departments within a big company or business partners within a virtual corporation. From the customer's point of view, its software is unlikely to be abused as it is (as a rule) only a small part of a distributed application. With distributed systems like DCE, data transfer between different components of a distributed application can also be encrypted. A more secure internet and more secure operating systems in the future might help to assert ReGTime-based concepts even in the envisioned worldwide open computing power market.

Our next aim is to test ReGTime in industrial projects. Therefore, we have to explore the integration of ReGTime into the existing electronic market and the way to originate the mentioned computing power market. We will examine which economical principles have to be considered, such that both customers and providers are willing to accept this new market.

References

[1] Larry Smarr and Charles E. Catlett. Metacomputing. *Communications of the ACM*, 35:45–52, June 1992.

[2] Friedhelm Ramme. Building a virtual machine-room — a focal point in metacomputing. *Future Generation Computer Systems*, 11:477–489, 1995.

[3] Ian Foster. High Performance Distributed Computing: The I-WAY Experiment and Beyond. In *Euro-Par96*, pages 3–10, 1996.

[4] D. Tavangarian. Hypercomputing mit Workstation-Clustern. http://www.tec.informatik.uni-rostock.de/hypercomp/, 1996.

[5] Rechenzentrum Karlsruhe. Virtuelles Rechenzentrum Karlsruhe. http://www.uni-karlsruhe.de/Uni/RZ/VRZ/, 1996.

[6] M. Weber and E. Nagel. Metacomputing zwischen KFA und GMD: Ein verteilter massiv-paralleler Rechner. *PARS-Mitteilungen*, 14:9–18, December 1995.

[7] Andrew S. Grimshaw et al. The legion vision of a worldwide virtual computer. *Communications of the ACM*, 40(1):39–45, January 1997.

[8] R. Baraglia, G. Faieta, M. Formica, D. Laforenza, and M. Stiavelli. Metacomputing to solve complex computational problems. In *Conference on*

High-Performance Computing and Networking, pages 429–440. Springer Verlag, Lecture Notes in Computer Science, Vol. 1067, April 1996.

[9] V.S. Sunderam, A. Geist, J. Dongarra, and R. Mancheck. The PVM concurrent computing system: Evolution, experiences, and trends. *Parallel Computing*, 20:531–546, April 1994.

[10] Message Passing Interface Forum. *MPI: A Message Passing Interface Standard*, June 1995.

[11] S. Müller, K. Müller-Jones, W. Lamersdorf, and T. Tu. Global Trader Cooperation in Open Service Markets. In *Trends in Distributed Systems*, pages 214–227. Springer Verlag, Lecture Notes in Computer Science, Vol. 1161, 1996.

[12] A. Puder and C. Burger. New Concepts for Qualitative Trader Cooperation. In *Distributed Platforms*, pages 301–313. Chapman & Hall, 1996.

[13] M. Merz, K. Müller-Jones, and W. Lamersdorf. Agents, services, and electronic markets: how do they integrate? In *Distributed Platforms*, pages 287–300. Chapman & Hall, 1996.

[14] Stefan Zbornik. *Elektronische Märkte, elektronische Hierarchien und elektronische Netzwerke*. Gesellschaft für angewandte Informationswissenschaft Konstanz (GAIK) e. V., 1996.

[15] NTIA Office of Assistant Secretary. Electronic commerce. http://www.ntia.doc.gov:80/opadhome/ecom3.hml, December 1995.

[16] M. B. Sarkar, B. Butler, and C. Steinfield. Intermediaries and cybermediaries: A continuing role for mediating players in the electronic marketplace. *Journal of Computer-Mediated Communication*, 1, 1996.

[17] Craig Milo Rogers, Anna-Lena Neches, and Paul Postel. Transitioning to the web. In *Online Procurement at ISI*. World Wide Web Fall Conference, 1994.

[18] Andersen Consulting. BargainFinder Agent. http://bf.cstar.ac.com/bf/, 1996.

[19] A. Brüggemann-Klein, G. Cyranek, and A. Endres. Die fachlichen Informations- und Publikationsdienste der Zukunft — eine Initiative der Gesellschaft für Informatik. pages 2–12. GISI, 1995.

[20] Ward Rosenberry, David Kenney, and Gerry Fisher. *Understanding DCE*. O'Reilly & Associates, Inc., 103 Morris Street, Suite A, Sebastopol CA 95472, September 1992.

[21] D. E. O'Leary, D. Kuokka, and R. Plant. Artificial intelligence and virtual organizations. *Communications of the ACM*, 40(1):52–59, January 1997.

[22] H. R. Hansen. *Klare Sicht am Info-Highway*. Verlag Orac, Wien, 1996.

[23] Simson Garfinkel. *PGP: Pretty Good Privacy*. O'Reilly & Associates, Sebastopol, CA, March 1995.

[24] M. Mütze. Firewalls. *Wirtschaftsinformatik*, pages 625–628, December 1996.

[25] Larry Wall and Randal L. Schwartz. *Programming Perl.* O'Reilly & Associates, Inc., Sebastopol, CA, 1991.

[26] B. Dreier, A. Huber, H. Karl, T. Ungerer, and M. Zahn. ReGTime — Rent Gigaflops someTimes. In *Trends in Distributed Systems*, pages 84–93. Springer Verlag: Lecture Notes in Computer Science, Vol. 1161, 1996.