A PROCESS ALGEBRA APPROACH TO GRID PARALLELISM.

MUFFY CALDER, PAUL COCKSHOTT, LEWIS MACKENZIE, MOHAMED OULD-KHAOUA, VIKTOR YARMOLENKO

Muffy Calder is a Professor of Computing Science at the University Glasgow, where she is Head of Department. She has extensive experience of developing and applying formal modelling and analysis techniques to a variety of real-world problems, particularly those involving concurrency. Her recent research is focussed on protocol analysis [5], feature interactions and advanced telecommunications services [2,3], stochastic process algebra for biochemical systems [1], symmetry and induction for model checking [4], and modal logics for process algebras [6]. She has experience of a number of model-checkers, with a particular interest in the Spin model checker [7].

She has been a principal investigator on 10 EPSRC funded projects and was co-chair of the *Sixth International Conference on Feature Interactions*, 2000. She is currently a principal investigator on a major DTI funded project: the (£1.1M) Beacon Project "A Software Tool for Simulation and Analysis of Biochemical Networks" (under the *Harnessing Genomics Programme*), in colloboration with Beatson Cancer Research Institute. In addition to developing new modelling techniques, this project involves probabilistic model checking on a Grid of 90+ machines. She has over 10 years experience of collaboration with telecommunications companies and was the principal investigator on the recently completed EPSRC research project "Veriscope – Verification of Similar, Concurrent Processes" (GR/R39/22/01).

Dr Paul Cockshott is a Reader in Computing Science and also has a background in economics with a strong research record in the areas of programming languages, video-compression and 3D computer imaging. In his current post, he has been investigating accelerated 3D imaging using multi-media instruction sets [8,9] and Grid Parallelism [10] with the Faraday Partnership and the former Turing Institute. His previous work has contributed to UK competitiveness through the development of the Strathclyde Compression Transform [11], a UK and world first for video-telephony over mobile GSM links. He has published widely on his research, including five books published and two in press. Dr Cockshott was co-investigator for SERC/EPSRC grants GR/F35881 ”Parallel Databases”, GR/J92170 ”Compressed Databases” and GR/J 07082 ”Space Machine”. He is currently principal investigator on GR/M54070 ”Real time 3D modelling”. He was co-supervising SHEFC RDG 103 ”Michelangelo Project” and is principal investigator for a DERA grant ”3D whole-body measurement system”, a Scottish Enterprise "Proof of Concept" Award on 3D microscopy image compression [12], and most recently, along with Lewis Mackenzie, was awarded a National Escience Centre grant to use Grid Parallelism in 3D vision.

Dr Lewis Mackenzie is a Senior Lecturer at the Department of Computing Science, University of Glasgow. His research interests are in computer architecture, parallel computing, networks and simulation. From 1989 to 1994 he was the principal researcher on the COBRA project, which was funded by Motorola Ltd. The aim of this work was performance modelling of networks in large scalable multi-computers, supporting up to thousands of processors. COBRA particularly focussed on the hyper-mesh, which is particularly suitable for the design of systems with high-volume fine-grain traffic, including systems with distributed virtual shared memory. He was also a principal researcher on FASTRAK, a project to design hardware to support a two dimensional hyper-mesh network based on LAN nodes. Dr Mackenzie has published extensively in the area of performance modelling of interconnection networks with numerous studies into the effectiveness of routing mechanisms in various network topologies subjected to different traffic patterns and conditions. He was co-chair of UKPEW 2002 and has been a member of a number of international program committees. He was the Guest Editor of a Special Issue on Performance Evaluation: Techniques & Tools in IEE Proceedings: Computers & Digital Techniques. He and Dr Cockshott are currently principal investigators on the NeSC-funded PGPGrid project [14].

Dr Mohamed Ould-Khaoua is a Reader in the Department of Computing Science, University of Glasgow. He has been working in the area of performance modelling/evaluation for over 13 years. He has extensively used analytical and software simulation models to investigate the performance of supercomputer architectures and networks. He is one of the leading international researchers on performance modelling of wormhole-routed networks. He proposed the first analytical model of adaptive wormhole routing in regular networks [16], and important traffic patterns [15]. He has also been actively researching other areas such as interconnection networks, parallel algorithms, fault-tolerant routing, and object replication. Dr. Ould-Khaoua has published extensively in the literature; over 65 papers in archival journals and over

100 papers in conference proceedings. He was the founding co-chair of the Int. Workshop series on performance modelling, evaluation, and optimisation of parallel and distributed systems (PMEO-PDS) held in conjunction with ACM/IEEE-IPDPS. He was the co-chair of UKPEW'02. He is the Guest Co-Editor of a number of special issues on systems and networks performance evaluation in the journal of Performance Evaluation, Computation and Concurrency: Practice and Experience, IEE Proceedings: Computers & Digital Techniques, Supercomputing, International Journal of High Performance Computing and Networking, and Journal of Parallel & Distributed Computing. He has served on the program committees of over 36 international conferences/workshops, including MASCOTS, IPCCC, SPECTS, ICCCN, MSWIM, and MOBIWAC.

Dr Viktor Yarmolenko has worked in the area of computational physics for 5 years and in area of parallel computing for over a year. His current interests lie in the area of computing to solve problems in science and industry. He started his career as experimental physicist while his undegraduate studies, which resulted in publication [17]. Then he worked on the computer simulation of liquid crystals in Sheffield Hallam University for 4 years, funded by QinetiQ. In his work Dr Yarmolenko used a wide range of computational techniques, e.g. Monte Carlo sampling, Fourier transform and alike, statistical computation, etc. Later on he worked as a digital video engineer for Chameleon Television Ltd for eight months and then joined Glasgow University (GU) to work on parallelisation of 3D stereographic matching algorithms for a full body colour 3D video scanner, developed by Faraday Group at GU, a NeSC funded project [10]. His work produced a novel parallel architecture [14], based on π-calculus [13], for highly asymmetric dynamic problems.

The applicants All applicatns are currently affiliated with the Department of Computing Science of the University of Glasgow. Dr Cockshott and Dr Mackenzie are current supervisors of Dr Yarmolenko on the PGP Grid Project [10].

The Department is rated 5 in the Research Assessment Exercise, and has an excellent reputation for the quality of its research. The Department is well founded with excellent computer and infra-structural support.

1. Executive Summary

The aim of this project is to investigate a framework for Grid computing based on a combination of process algebra and market mechanisms. We aim to develop:

- A π calculus based mechanism for Grid parallelism.
- A mechanism by which an audit trail can be created for work undertaken and contractual proof of the performance of computing services.
- A resource location and brokerage facility tied to a price mechanmism for the allocation of jobs.
- Protocols for the negotiation of contracts for computational jobs, and formal verification of their adequacy.

The framework for the computations will be Java, and we will develop a Java API to support this. We will address the problem of fraudulent claims that computing work has been done by developing appropriate forms of digital signatures to proove that work has been performed as contracted. We will develop proofs of the correctness of the communications protocols used to implement the π calculus over the Grid and investigate the security of the computational contracts in the face of malicious intent.

The project brings together an experienced team with expertise in high performance computing, simulation, Grid computing, protocol analysis, formal modelling and automated reasoning. We will build upon prior work by the team in developing a large scale applications for image processing and animation, and a prototype Grid implementation of the π calculus in Java [14]. We will employ model-checking techniques for reasoning about the Java π calculus.

The novelty of the project is in the way we combine and explore the concepts of labour exchanges, peer-to-peer protocols between labour exchanges and servants, and contractual proof that a computation has been carried out.

The project will run for 2years and employ 2 Postdoctoral RAs.

2. Introduction

Although the essence of the Grid is the harnessing of parallelism in the large, the practical tools used in the Grid community for handling parallelism have grown up largely independently of the formal models for parallelism developed by the theoretical computer science community. Existing Grid [18] protocols have been developed in the context of government or academic computing centres. In such centres access is usually based on quotas of computer time, allowing certified users to access super-computer resources. The characteristic of computer centres is that most of the jobs are based on batch processing, which diminishes the greatest potential of a Grid, that is dynamic resource discovery. Another characteristic of computer centres is that they are typically behind firewalls that are politically hard to reconfigure. This tends to limit Grid protocols to a few ports that are guaranteed to be visible through the firewalls.

The spread of broadband internet over the last 3 years means that a huge new computing resource has come onto the internet. In terms of CPU power, this resource is substantially greater than the resources of super-computer centres. The processors in home machines are competitive, exist in large numbers, and now have communication links that make them available for a sufficiently long time. This has allowed the proliferation of peer-to-peer computing protocols for file sharing, instant messaging, VoIP, Video Streaming, *etc*. These protocols offer an immediate benefit to the users, are easy to set up and require no heavyweight certification.

This proposal will explore alternative routes to Grid parallelism based on a combination of process algebra and market mechanisms. We suggest that process algebra provides a theoretically sound framework for the structuring of Grid parallelism, and that the incentive structure of the market provides a socially distributed mechanism to call forth additional computational resources. In recent years, various attempts have been made to exploit private remote computing resources [21, 23, 36, 37]. Several of these have been in the context of predefined parallel tasks [25–29], most famously the SETI@home project [21]. Others progressed further in that the screen saver code did not consist of a dedicated scientific application, but new task could be uploaded [30, 31]. Its limitations were non dynamic scheduling, not Grid enabled, tasks could not spawn other tasks and could not communicate with each other, job submission was centralised. Currently they are working on a market mechanism [32, 33] similar to the CX Project [34]. CX is a Jini-Based Computation Exchange system in which arbitrary java tasks are submitted by clients to a market where they are picked up to be processed.

The work flow paradigm in these systems is centralised. Tasks do not communicate directly with one another and data flows only between the initiator and the servant machines. This contrasts sharply with the model of parallelism suggested by the π calculus [13]. This not only allows tasks to communicate, it allows the data-flow graph itself to be dynamically changed during the process of computation. This is a more powerfull model of parallelism than others such as [23, 36, 37], and is much better suited to the needs of applications like image processing, computer vision and computer animation. All these can involve streaming image data through a processing graph whose structure may need to alter to accomodate the data.

Our aim is to explore the challenge of using the π calculus paradigm within the context of a distributed and hetrogenously owned pool of computers. We intend to build upon prior work by the team in developing a prototype Grid implementation of the π calculus in Java [14].

3. Research Program

We have identified 5 major problems which must be addressed in order to use CPU cycles across hetrogenously owned machines:

(1) How can one deal with the variety of computers that exist on the internet, allowing a job to be specified without any prior knowledge of the operating system or processor on which it will run.
(2) How can the person/organisation offering CPU cycles be sure that no malicious actions will be performed by the program that runs on their computer.
(3) How do you find computers ‘willing’ to do your job.
(4) How is payment made for work done.
(5) How can proof be produced that the work has been done.

Progress has already been made on the first two problms by existing Java technology and the use of the π calculus paradigm. For example, Java already provides a compile once, run anywhere, paradigm through its portable byte-codes [42]. The Java ’sandbox’ allows alien code to run on a users machine in whilst protecting it from memory over-runs and excluding any malicious access to the local file-space. Its incorporation into web browsers means that running alien Java code on client machines is commonplace. And since the π calculus paradigm assumes all data accessed by a task arrives on transmissible communications channels, such computations need never access local filespace. All data needed for the computation will be sent over the internet from the originating process. All results will be sent via memory-to-memory transfer, without accessing local file store. We can therefore concentrate on solving problems of resource discovery,accounting, payment and audit trails for compute tasks.

We propose to employ three crucial concepts, combined in a novel way: labour exchanges, peer-to-peer protocols between labour exchanges and servants, and contractual proof that a computation has been carried out. These are outlined below.

P2P. We propose to use a peer-to-peer protocol with servers analogous to systems like Gnutella [19]. Machines are divided along Gnutella lines into two categories labour exchanges (LE) and servants (SV). SV can both initiate and accept compute jobs on the network. The entities involved in our model are Abstract Juridical Personalities, (AJP), LE and SV. An AJP is either a person, a company or any other institution capable of pursuing a claim in a court of law. A LE is a computer with linkage to the banking system capable of triggering payments between AJPs. A SV is a computer owned by an AJP that is made available to trade CPU cycles. It acts as a Robot for hire. The term Robot which entered English from the Czech play RUR [41], had a prior meaning, among other concepts relating to work, of labour services owed by a peasant to a feudal lord or Boyar. Rossum's Robots were his mechanical slaves. Our computers stand in a similar relationship to us.

Labour Exchange. To deal with problems 3 and 4 we use the concept of Labour Exchange. These embody both the administrative notion of a labour exchange in the British social security system as a place to which you report to find labour when you are unemployed, and also Robert Owen's idea of labour exchanges as places where citizens' mutual labour debts would be cancelled [43]. These notions derived from public policy are given physical embodiment in a server similar to those used in the Gnutella network. Since the computational tasks will be distinct and run on computers of different powers with different instructionsets we need some means of equating work on these various machines. This is analogous to the problem of reducing complex to simple labour in value theory. We propose to quote all job costs in standard Java cpu minutes, equating the performance of different machines using a representative Java instruction mix. The costs in Java cpu minutes would be converted to money prior to final settlement using a Monetary Equivalent of Computer Labour.

Contracts. To deal with problem 5 we use notions from Turing [44] and from public key cryptography [45]. Turing posited both a model of computation, the Turing Machine, and a generalisation, the so-called 'Universal Turing Machine' (UTM). All Turing Machines are finite automata augmented by one or more infinite read/write sequential stores or 'tapes'. The UTM is one which when given an input tape made up of the sequence uip will emulate the behaviour of any other Turing machine i applied to input data p The Java Virtual Machine is computationally equivalent to a bounded UTM. When provided with an input consisting of a jar file u_j followed by a sequential data stream p it will run the j-th possible Java program on data p. The set of possible Java programs can be enumerated by treating their jar file encodings as binary integers. Since any finite bounded Turing Machine can be emulated as a Java program working on an instance of the Vector class, it follows $\forall u_i \exists u_j$.

Suppose we run a specific Turing Machine ui on input p_k then, provided the machine halts, we are left with an answer a_k that was fully specified by $u_i p_k$. The answer was implicit in the specification of the problem but computational work was required to produce it. It is this computational work that we propose to trade. Suppose that when machine A agrees to do work for machine B, they record a sealed version of $u_i p_k$ with a trusted third party E. When A completes the task and sends a_k to B, it also sends a copy to E. A has then registered proof with E that the contracted computational task has been done. If B disputes payment, the sealed copy of $u_i p_k$ can be run by E on a reference UTM to see if the answer is a_k. We will encode the results as follows. It is common usage in file sharing networks to use secure hashes to identify files on servers in a way that is independent of their file names [46]. A similar approach can be used to the pairs $u_i p_k$ and the answers a_k, so that instead of recording the entire program, data and answer at the server, a secure hash of the data, program and answer are stored. With sufficiently long hashes the chances of misidentifying programs data and answers are vanishingly small.

4. Beneficiaries

The development of a market in CPU time would have wide economic implications. It would open up a new area to commercialisation in a manner analogous to the creation of the Web. One can envisage that it would result in the formation of new forms of companies specialising to meet this market. Just as the web created a market for companies supplying server farms, one can envisage similar opportunism opening up to create commercial processor farms. The likely initial beneficiaries of this would be the firms who currently operate server farms.

The additional network traffic created by the distribution of processing resources would be to the benefit of Telecoms companies and ISPs.

We envisage that there would be market opportunismin the operation of Labour Exchanges, which would operate in a fashion similar to banks in the clearing of credit and debit balances for processing work done, and in converting these balancesinto 'real' money. It would be reasonable to expect that other companies would come into existence specialising in the provision of intermediary services - bulk discounted processing time and the like.

Firms which have occasional bursty requirements for massive processing power - for example the film industry who need large quantities of render time for a particular film - would be saved the capital overhead of having to buy farms of servers themselves.

5. WORKPACKAGES

5.1. **Registration.** Development of a mechanism which would allow servants to register the resources and QoS they are able and willing to provide with central Labour Exchange servers.

5.2. **Market Mechanism.** Specification of protocols using the approach of Section 1 which would allow a market to operate between systems requiring work to be done (employers) and servants using Labour Exchanges as enablers.

5.3. **Job Invocation.** Development of job invocation subsystem which would include querying the LA server. This would be based on the JPie system developed under the PGPGrid project [10].

5.4. **Secure blocking communications.** Development of a second generation communication sub-system for our Java πcalculus that is both cryptographically secure and fully implements the π calculus blocking protocols.

5.5. **Audit Trail.** Establishing mechanisms which would allow verification of contractual claims that work has been done by a servant. We intend to use the legal concept of escrow in conjunction with secure hashes on the problem specification and on the answer to provide independently verifiable evidence both of contracts and their performance.

5.6. **Implementation of Labour Exchange.** This will be achieved using web service technology developed from earlier work accomplished during the PGPGrid project [10], currently being undertaken by 2 of the proposers in conjunction with EPCC.

5.7. **Experimental Testing.** Construction of an experimental system of modest size (¡100 machines) to test the operation of the mechanisms in practice. It is anticipated that this will use existing resources within the department.

5.8. **Consistency of two way Contracts.** Construction of proofs of the consistency of the protocols outlined in 3. We will use the model checker SPIN [7], and its associated source language Promela, for modelling these protocols. Promela has both synchronous and asynchronous communciation, and dynamic process communication, making it ideally suited to protocol analysis. Verifications are expressed by linear temporal logic formulae.

5.9. **Consistency of implementation of π calculus communication implementation.** Construction of formal proofs that the Java implementation of the communications protocols accurately reflects the semantics of the original π calculus. Again we will use the model checker SPIN [7], and its associated source language Promela, for modelling and reasoning. Promela is ideally suited for this task because it is one of the few specification languages which includes dynamic communication channels (i.e. channels on channels). Thus we can model the π calculus elegantly and efficiently.

5.10. **Consistency of multi-way contracts.** Extending the methods of 3 and 5.8 to handle the multi-way communication presupposed by the π calculus. We regard it as a scientifically open question as to whether there exist computationally tractable protocols that would achieve this given the computational universality of the π calculus itself.

6. Personnel, Management and Workpackage Scheduling

To execute these workpackages we require two RAs for 2 years who will investigate specific aspects of the overall programme. The work will be organised as follows. The two RAs will initially develop technologies necessary to implement a prototype labour exchange. RA1 will focus on extending the ideas generated by the PGPGrid project, including the JPie system, and RA2 on the market mechanisms and audit trail technologies required to allow a realistic market, of the type proposed, to function.

After this initial phase which we estimate will take just under a year, RA1 will be responsible for the implementation and experimental testing of the prototype LE and the subsequent reporting of results. The main thrust of RA2s work however will shift to constructing consistency proofs of the contractual and *pi* calculus communication protocols.

Overall project management will be the responsibility of Dr. Cockshott. Professor Calder will be responsible for directing the consistency proofs of the contractual and π calculus communication protocols. All the PIs and research assistants will meet on a fortnightly basis, to review overall project progress.

We are asking for 15% of the time of a computing officer, since the intention is to mount the π calculus parallelism framework accross existing labs of machines. For this we will need effort from staff responsible for maintaining the software on these laboratories.

The table below indicates the division of responsibilities between the two RAs in terms of workpackages and the estimated duration of each such package in months.

Workpackage	RA1	RA2
Registration	3	
Market Mechanism		5
Job Invocation	4	
Secure communications	4	
Audit Trail		5
Implementation of LE	6	
Experimental Testing	4	
Consistency of 2-way contracts		4
Consistency of communications		4
Consistency of m-way contracts		3
Reporting Results	3	3
	24	24

TABLE 1. The work plan.

7. Equipment

The following equipment is requested to support the personnel who will be engaged on the project.

(1) Two workstations to support the RAs in general development. These machines should also be capable of generating work for the RUR system to handle in order to provide stress loading of any experimental system. An approximate specifications as follows should suffice: 3GHz Pentium 4 with 2 Gbytes of RAM, 2 x 120Gbyte disks, standard graphics, TFT screen. Estimated cost of such systems at current market prices is approximately 1200 each (inc VAT).
(2) A server will be required to host the experimental labour exchanges. The machine should be powerful enough to virtualise the exchanges allowing us to emulate the effect of operating multiple work distribution nodes. We are interested in examining the behaviour of 64-bit implementations and so an appropriate system would be as follows. A dual Opteron server with 8Gbytes of RAM and 2x250Gbyte drives. Estimated cost 3000 (inc VAT).
(3) A local unmanaged 24-port gigabit Ethernet switch. Approximate cost 500.

The work will make extensive use of a number of existing facilities, including 20 local high-end systems (15 of which are dual processor configurations) to which we have negotiated limited access for work-farm experimentation. Additionally, the 3Dmatic laboratory will allow us to use the 3D-body scanner system as a genuine application from our target set which to generate real high-volume workloads (up to 200 tasks per second) to impose loading stress on the experimental RUR system.

8. DISSEMINATION AND EXPLOITATION

As the project progresses, outputs from this research will be disseminated in the usual way, through journal publications, conference and workshop presentations. The primary contenders are 'International Journal of High Performance and networking', 'IEEE Parallel & Distributed Technology', and a selection of conferences from ASE, ICSE, IJCAI, PODS, ECAI, ETAPS, MFCS, CSL, CAV, LICS, TFME, the Grid "All Hands" meetings, Global Grid Forums, and workshops on verification and Java.

At the same time, we may seek industrial sponsorship for a spin off.

REFERENCES

[1] M. Calder, S. Gilmore, J. Hillston. Modelling the influence of RKIP on the ERK signalling pathway using the stochastic process algebra PEPA. In Proceedings of Bio-Concur 2004, *Electronic Notes in Theoretical Computer Science*, 2004.

[2] M. Calder and E. Magill. *Feature Interactions in Telecommunications and Software Systems VI*, IOS Press 2000. ISBN 1 58603 065 5.

[3] M. Calder and A. Miller. Using SPIN for Feature Interaction Analysis - a case study. In Proc. 8th Int. SPIN Workshop (SPIN 2001), Toronto, Canada, *Lecture Notes in Computer Science*, Springer, vol. 2057, pp. 143–162, 2001.

[4] M. Calder and A. Miller. Automated verifi cation of any number of concurrent, communicating processes. *Proc.* 17^{th} *IEEE Conf. on Automated Software Engineering (ASE 2002)*, pp. 227-230, *IEEE*, September 2002.

[5] M. Calder and A. Miller. Using SPIN to Analyse the FireWire protocol - A Case Study. *Formal Aspects of Computing Science (FACS)*, volume 14(3), pages 247–266. Springer-Verlag, 2003.2003.

[6] M. Calder, C. Shankland and S. Maharaj. A Modal Logic for Full LOTOS based on Symbolic Transition Systems. *The Computer Journal*, vol. 45, no.1, 2002.

[7] Gerard J. Holzmann. *The SPIN model checker: primer and reference manual*. Addison Wesley, Boston, 2003.

[8] Direct Compilation of High Level languages for Multimedia Instruction Sets, Technical Report, *Department of Computing Science*, Univeristy of Glasgow, Ocrober (2000)

[9] P. Cockshott, *Vector Pascal reference manual*, Sigplan Notices, **37**, issue 6, p59

[10] Peppers Ghost Grid Project (#0000), http://www.epcc.ed.ac.uk/pgpGrid/ (*http://www.peppersghost.co.uk/*) GU is currently engaged in a collaborating project with PGP, funded by NeSc.

[11] P. Cockshott, R Lambert, *Algorythm for the Hierarchical Vector Quantization of Video Data*, IEE Procedings Vision, Image and Signal Processing, August (1999)

[12] P. Cockshott, Y. Tao, G. Gao, P. Balch, A.M. Briones, C. Daly, *Confocal Microscopic Image Sequence Compression Using Vector Quantization and 3D Pyramids*, The Journal of Scanning Mictoscopies, AAAI Press, p8

[13] R. Milner, *The Polyadic* π*-calculus, a tutorial* (1991)

[14] P. Cockshott, L. Mackenzie, V. Yarmolenko, *PGP Grid Project Proceedings to 10th Global Grid Forum*, (Mar 2004), *http://www.epcc.ed.ac.uk/pgpGrid/docs/PGPGridGGF10final.pdf*

[15] M. Ould-Khaoua, "A performance model of Duato's fully-adaptive routing in k-ary n-cubes", *IEEE Transactions on Computers*, **48**, # 12, 1-8, (1999)

[16] M. Ould-Khaoua, H. Sarbazi-Azad, "An analytical model of adaptive wormhole routing in hypercubes in the presence of hot-spot traffi c", *IEEE Transactions on Parallel and Distributed Systems*, **12**, # 3, 283-288 (2001)

[17] O. Kovalchuk, V. Reshetnyak, V. Yarmolenko, O. Yaroshchuk, *Molecular Crystals and Liquid Crystals*, **Sec. A, IV**, 2487 (1998)

[18] Global Grid Forum, *http://www.ggf.org*

[19] Gnutella fi le sharing system, *http://www.gnutella.com*

[20] Nelson Minar *et al*, *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*, O'Reilly (Mar 2001)

[21] ET Search project: *http://setiathome.ssl.berkeley.edu/unix.html*

[22] C. Sharp, *Business Integration for Games: An introduction to online games and e-business infrastructure*, IBM white paper (2003), can be found on *http://www-106.ibm.com/developerworks/webservices/library/ws-intgame/*

[23] Message Passing Interface White paper, documents are available on *http://www.mpi-forum.org*

[24] BBC News Online: Sci/Tech. Tuesday, August 17, *http://news.bbc.co.uk/1/low/sci/tech/423022.stm*

[25] *http://www.npaci.edu/online/v6.14/lifemapper.html*

[26] *http://www.computeagainstcancer.org/*

[27] *http://www.amarbio.com/*

[28] AIDS@home, Folding@home, *etc.*, *http://www.scientopica.com/resources/*

[29] *http://icosystem.com*

[30] Parabon Inc, The Frontier Application Programming Interface,*http://www.parabon.com/developers/frontier/FrontierAPI/index.html* 2004.

[31] W. L. George et al, "Accelerating Scientifi c Discovery Through Computation and Visualization II", *NIST Journal of Research*, **107**(3), 223-245 (May-Jun 2002), *http://math.nist.gov/mcsd/savg/parallel/screen/*

[32] R. F. Brooks, I. N. Lings, M. A. Botschen, *The Service Industries Journal*, London (Oct 1999);

[33] Van Haastrecht, R. and M. Bekkers, *24th EMAC Conference Proceedings*, 1243 (1995)

[34] Peter Cappello, Dimitrios Mourloukos, *A Scalable, Robust Network for Parallel Computing* , Joint ACM Java-Grande - ISCOPE 2001 Conference, pp. 78 - 86, Stanford University, California, June 2 - 4, 2001

[35] A. Baratloo, M. Karaul, Z. Kedem and P. Wyckoff, ”Charlotte: Metacomputing on the Web”, *Proceedings of the 9th Conference on Parallel and Distributed Computing Systems* (1996)

[36] W. Grosso, *Java RMI*, O’Reilly, 2001.

[37] Al Geist, Adam Beguelin, Jack Dongarra, Weicheng Jiang, Robert Manchek, Vaidyalingam S. Sunderam *PVM: Parallel Virtual Machine - A Users’ Guide and Tutorial for Networked Parallel Computing (Scientific & Engineering Computation S.)* ,MIT Press, 1995.

[38] Peter Welch, *CSP Networking for Java (JCSP.net)*,Global and Collaborative Computing Workshop, ICCS 2002, Amsterdam (22nd. April, 2002)

[39] Christopher F. Joerg, ”The Cilk System for Parallel Multithreaded Computing”, *Ph. D. Thesis, MIT Dep-t. of EECS* (Jan1996), see also *http://supertech.lcs.mit.edu/cilk/index.html*

[40] R. D. Blumofe, C. F. Joerg, B. C. Kuszmaul, C. E. Leiserson, K. H. Randall, and Y. Zhou, ”Cilk: An Efficient Multithreaded Runtime System”, *Proc. 5th ACM Symposium on Principles and Practice of Parallel Programming*, Santa Barbara (Jul 1995)

[41] English publication of ”Rossum’s Universal Robots” in Karel Capek’s *Four Plays* (1999)

[42] D. Kramer, ”The JavaTM Platform: A White Paper”, with contributions by B. Joy, D. Spenhoff (May 1996), see also *http://java.sun.com/docs/white/platform/javaplatform.doc1.html*

[43] Robert Owen, A new view of society and report to the County of Lanark, Penguin 1970, first published (1813)

[44] Alan Turing, ”On computable numbers, with an application to the Entscheidungsproblem”, *Proceedings of the London Mathematical Society*, Ser. 2, **42** (1937)

[45] R. Rivest, A. Shamir and L. Adleman, ”A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”, *Communications of the ACM*, **21** (2), 120-126 (Feb 1978)

[46] Ian Clarke, Oskar Sandberg,Brandon Wiley,Theodore W. Hong, ”Freenet: A Distributed Anonymous Information Storage and Retrieval System” in *Designing Privacy Enhancing Technologies*: International Workshop on Design Issues in Anonymity and Unobservability, LNCS 2009, ed. by H. Federrath. Springer: New York (2001).