A problem-solving environment (PSE) for distributed computing

Shigeo Kawata*

Department of Electrical and Electronic Engineering, Graduate School of Engineering, University of Utsunomiya, 7–1–2 Yohtoh, Utsunomiya 321–8585, Japan Fax: +81-28-6896081 E-mail: kwt@cc.utsunomiya-u.ac.jp *Corresponding author

Hitohide Usami

National Institute of Informatics, Center for Grid Research and Development, Jinbocho Mitsui-building 14F(NAREGI), 1-105, Kanda-Jinbocho, Chiyoda-ku, Tokyo 101-0051, Japan Fax: +81-03-4212-2803 E-mail: usami@grid.nii.ac.jp

Yoshikazu Hayase

Department of Electronic Control Engineering, Toyama National College of Maritime Technology, Shinminato 933-0293, Japan Fax: +81-766-86-5110 E-mail: hayase@toyama-cmt.ac.jp

Yutaka Miyahara, Motohiro Yamada and Masahide Fujisaki

Computational Science and Engineering Center, Fujitsu Limited, Mihama-ku, Chiba 261-8588, Japan

Yohji Numata, Shinji Nakamura and Noriyuki Ohi

Department of R&D Solution, Fujitsu Nagano Systems Engineering Limited, Nagano 1403-3, Japan Fax: +81-26-237-4115 E-mail: numa@nagano.fujitsu.com E-mail: snaka@nagano.fujitsu.com E-mail: ohi@nagano.fujitsu.com

Masami Matsumoto

Department of Electrical Engineering, Yonago National College of Technology, 4448 Hikona-cho, Yonago City, Tottori 683-8502, Japan Fax: +81-0859-24-5049 E-mail: matsu@yonago-k.ac.jp

Takayuki Teramoto

Department of Electronics and Computer Engineering, Tsuyama National College of Technology, 624-1, Numa, Tsuyama 708-8509, Okayama, Japan Fax: +81-868-24-8247 E-mail: teramoto@tsuyama-ct.ac.jp

Masumi Inaba, Ryouji Kitamuki and Hideaky Fuju

Graduate School of Electrical and Electronic Engineering, University of Utsunomiya, 321–8585, Utsunomiya, Japan

Yasuhiro Senda and Yoshio Tago

Department of Computational Science, University of Kanazawa, Kakumal, Kanazawa 920-1192, Japan

Yukio Umetani

Department of Computer Science, University of Shizuoka, 3-5-1 Jyouhoku, Hamamatsu 432-8011, Japan Fax: +81-53-478-1452 E-mail: umetani@cs.inf.shizuoka.ac.jp

Abstract: A problem-solving environment (PSE) for a distributed high-performance computing (HPC) is proposed to help users to work on distributed computer environment. When users access and use distributed computers for scientific computations, the PSE tells users which computers are available and appropriate for their specific application software by using hardware and software informations specified. Then, the users deploy their software on the distributed computer systems. The software, which can be used later or available for other users, is plugged and pooled in the PSE application pool in order to enhance their reusability. The problem-solving process work flow (WF) is also stored in a case-based database (DB) in the PSE and then the case DB can be used to suggest a WF for users to solve their problems. The PSE may open a new flexible HPC environment.

Keywords: computer-assistant computing; distributed computing; HPC; problem-solving environment.

Reference to this paper should be made as follows: Kawata, S., Usami, H., Hayase, Y., Miyahara, Y., Yamada, M., Fujisaki, M., Numata, Y., Nakamura, S., Ohi, N., Matsumoto, M., Teramoto, T., Inaba, M., Kitamuki, R., Fuju, H., Senda, Y., Tago, Y. and Umetani, Y. (2004) 'A problem-solving environment (PSE) for distributed computing', *Int. J. High Performance Computing and Networking*, Vol. 1, No. 4, pp.223–230.

Biographical notes: Shigeo Kawata graduated following a master course at the Graduate School of Science and Engineering, Tokyo Institute of Technology (TIT) in 1980. He received a PhD from TIT in 1985. He became a research associate in 1981 in TIT, in 1986 he moved to Nagaoka University as an Associate Professor and is now at Utsunomiya University (7-1-2 Yohtoh, 321-8585 Utsunomiya, Japan) as a Professor since 1999. He has worked on problemsolving environment (PSE), computational science and engineering and computational fluid and plasma simulations. He has been a member of IEEE, Japan Society of Computational Engineering and Science, Information Processing Society of Japan, American Physics Society and more.

Hitohide Usami and received his PhD from the Graduate School of Engineering, Tohoku University in 1978. He has worked for Fujitsu Limited since his graduation and temporary for the National Institute of Informatics for NAREGI project. His research interests are in knowledge-based systems, semantic grid and problem-solving environment (PSE). He has been a member of Japan Society of Computational Engineering and Science, Institute of Electronics, Information and Communication Engineering and Japan Society for Artificial Intelligence.

Yoshikazu Hayase graduated following the master course at the Graduate School, Nagaoka University of Technology. He is a research associate of Toyama National College (1-2 Ebie Neriya, 933-0293 Shinminato, Japan) since 1991. He has worked on computational science and engineering, computer education and problem-solving environment (PSE). He is a member of Japan Society of Computational Engineering and Science, the Japanese Society for Artificial Intelligence.

Yutaka Miyahara received a Bachelor of Philosophy from the University of Tokyo in 1990. He has worked on research and development of knowledge information processing, case-based reasoning and natural language processing. He is working on bio-informatics and grid computing.

Motohiro Yamada received a Master in Engineering from the Graduate School of Engineering, Iwate University in 1993. He entered Fujitsu Limited in 1993. He has worked on distributed/parallel computing, high-performance computing and grid computing.

Masahide Fujisaki received a Bachelor of Applied Mathematics from the Science University of Tokyo in 1984. He has worked on high-performance computing in the field of computational science, optimisation of nuclear codes, QCD codes and other scientific applications on parallel computer in special. He has worked on the integration of grid system development for scientific research since 2000.

Youjiro Numata is the section chief in Fujitsu Nagano Systems Engineering. He has worked on development of 'Science system for the laboratory organisation'.

Shinji Nakamura has developed 'Science system for the laboratory organisation' in Fujitsu Nagano Systems Engineering.

Noriyuki Ohi has developed 'Science system for the laboratory organisation' in Fujitsu Nagano Systems Engineering.

Masami Matsumoto graduated from the master course at the Graduate School of Engineering, Nagaoka University in 1988. He received a Dr Engineering from Tottori University in 2001. He became a research associate in Yonago National College of Technology (4448 Hikona-cho, Yonago City, Tottori Prefecture, 683-8502, Japan) in 1989 and became an Associate Professor in 1992. He has worked on computational science, electrical engineering, virtual reality and problem-solving environment (PSE). He has been a member of Japan Society of Computational Engineering and Science and the Virtual Reality Society of Japan.

Takayuki Teramoto graduated from the master course at the Graduate School of Electrical Engineering, Nagaoka University. He became a research associate in 2001 in Tsuyama National College of Technology. He has worked on computational science and engineering, computer education and problem-solving environment (PSE). He has been a member of Japan Society of Computational Engineering and Science, Information Processing Society of Japan, Japan Society for Artificial Intelligence and the Institute of Electrical Engineering of Japan.

Masumi Inaba is a student of the Master course of Department of Electrical and Electronic Engineering, Utsunomiya University.

Ryouji Kitamuki is a student of the Master course of Department of Electrical and Electronic Engineering, Utsunomiya University.

Hideaki Fuju is a student of the Master course of Department of Electrical and Electronic Engineering, Utsunomiya University.

Yasuhiro Senda graduated from the Department of Metallurgy, Tohoku University in 1993. He received a PhD from Hiroshima University in 2000. He became a research associate in 2000 in Kanazawa University. He has worked on computational physics, HPC, scientific visualisation and problem-solving environment (PSE). He is a member of Japan Society of Computational Engineering and Science and Physical Society of Japan.

Yoshio Tago has worked on PSE and Computings in science and engineering in Kanazawa University as a Professor.

Yukio Umetani graduated from the Department of Mathematics, Tokyo University in 1968. He worked on supercomputer architecture, vectoring compiler and numerical simulation system DEQSOL during his stay in Hitachi Ltd. from 1968 until 1996. He became a Professor at Shizuoka University in 1996 and has been conducting researches on numerical simulation and problem-solving environment (PSE). He is a member of Information Processing Society of Japan, Japan SIAM, Japan Society of Computational Engineering and Science, Acoustical Society of Japan and American Society of Computing Machinery.

1 Introduction

In high-performance computing (HPC) and simulations in science and technology, distributed computer systems including GRID systems (Global Grid Forum; Teramoto et al., 2001) are now popular and provide a flexible and cost-effective environment. On the other hand, the distributed computer system may be in a complex environment of heterogeneous computer hardware and software. However, it is quite difficult for users to know the whole structure of the distributed computer system they use. In a realistic and useful distributed environment, users should be supported by assistant software

(Global Grid Forum), so that users can use the distributed computer systems smoothly and effectively without the knowledge of distributed computer system itself.

In this paper, a new software of problem-solving environment (PSE) is proposed in order to support users who work on distributed computer systems. When users access and use distributed computers for scientific computations, the PSE server tells users which computers are available and appropriate for their specific application software by using hardware and software information specified. Then, the users deploy their software on the distributed computer systems. The software, which can be used later or available for other users, is plugged and pooled in the PSE application pool in order to enhance their reusability. The problem-solving process work flow (WF) is also stored in a case-based database (DB) in the PSE and later the case DB can be used to suggest a WF for users to solve their next problems. In addition, one of the PSE components provides a useful tool for parallel computations and it supports users to generate Message Passing Interface (MPI)-based parallel computer programs. The PSE may open a new flexible HPC environment in the HPC world.

In Section 2, the role of PSE in distributed computing is briefly presented, the PSE server is described in detail in Section 3 and an example application result is demonstrated in Section 4. Section 5 is devoted to summary.

2 Role of PSE in distributed computer system

Distributed computer system and GRID environment are now available for HPC and provides a new circumstance to utilise CPUs effectively and efficiently. In the distributed computers, users may find difficulties to utilise the computers distributed spatially. For users it may be quite difficult to know complete information of the hardware located apart from the users and software installed in the distributed system, though the information is required for the computations. In addition, users should also know job queue so that they can estimate when their jobs would finish.

In order to utilise the distributed computers and/or GRID, users require a software system to solve these difficulties. PSE serves a solution to overcome the difficulties. The definition of PSE is as follows: 'A system that provides all the computational facilities necessary to solve a target class of problems. It uses the language of the target class and users need not have specialised knowledge of the underlying hardware or software' (Gallopoulos et al., 1994). Therefore, there can be several kind of PSE to support users in various ways. The major aim of PSE research is to reduce time and effort required to solve target problems (CACTUS; Houstis et al., 2000; Kadooka et al., 2003; Konno et al., 1987; Schuchardt et al., 2002; Umetani et al., 1987; Umetani et al., 1991). In this paper, the PSE is a mediator between users and middleware, for example, Globus (Globus Toolkit), Unicore (Unicore Forum), etc., which supports a singlesign-on system, a secure communication, retrieval of information of hardware, software and job queue and a distributed resource management. The PSE communicates users directly to support them and solve the difficulties as shown in Figure 1. Without the PSE it is difficult to run users' application software on computers distributed geographically.

3 PSE server for HPC on distributed computers

3.1 Target issues of PSE server

The present research target issues of the PSE server include the followings: (1) simple and easy execution of users' application programs on distributed computers. (2) Simple and easy deploy of users' program onto the distributed computer environment. (3) Simple and easy software plug-in system into the application pool in the PSE server to provide a software-reuse environment. In this mechanism the PSE server supports users to reuse previous software for their own purposes. (4) Case database (Case DB) for problem-solving WF, based on that users can solve their problems. The WFs developed are also stored in the PSE server for the WF reuse. (5) Computer-assisted parallel program generation should be supported. For this purpose, a module called NCAS is also developed to support users to design and generate MPI-based parallel programs. (6) The PSE server should tell users where the computation results located in the distributed computer system.

On the basis of the PSE server presented in this paper, users may work smoothly on a distributed computer system: users need not to have precise information about hardware location behind the PSE server and execute their software without difficulties. The PSE server holds all the information about the distributed computer system and manages all the information relating to hardware, software installed, job execution, network and the location of computation results (Figure 2).

3.2 *PSE server structure*

The PSE server consists of three major components: PSE toolkit, PSE portal and application pool. The PSE portal serves a user interface to users, who work on distributed computer systems. The PSE toolkit provides core services to realise the target issues shown in the preceding subsection. The application pool stores application software registered by the PSE toolkit and developed or used by users.

Figure 1 Positioning of PSE in one job cycle in the distributed computer system or GRID. Users access the PSE to run their programs through the PSE



On the other hand, users design WFs to solve their problems on the PSE portal and the PSE toolkit manages the WF execution as shown in Figure 3. The application software required for the WF execution and stored in the application pool is deployed onto a distributed computer systems and the PSE toolkit presents the computers on Figure 2 PSE server for HPC on distributed computers. The PSE server consists of PSE toolkit, application pool and PSE portal. Users access the distributed computers through the PSE portal, and the PSE toolkit provides functions required for the execution: a WF is suggested by the case DB and the user reuses/modifies the WF. The user's software is plugged into the application pool, and deployed on the distributed computers. The computers are selected by the PSE toolkit and presented to the user. On the computers specified, the user application runs, and the PSE toolkit tells the user where the results are obtained



which the deployed application software runs. Before selecting the computer to use, the PSE toolkit shows a suggestion on computers among the distributed computers available. The PSE toolkit obtains requirements for the specific application software from the user and searches hardware on the network-linked computers. The case DB and the parallel program generator are included in the PSE toolkit for the users support.

3.3 PSE toolkit

In this subsection, the core component of the PSE toolkit is presented in detail. The PSE toolkit consists of the following subcomponents: (1) application and application information register, (2) a component to perform test compile and run at the register phase in order to check if the program can be compiled on the target CPUs, (3) application search component to find a correct program from the application pool, (4) application information retriever from users, (5) application deployment, (6) a component of WF search in and register at the application pool, (7) a component for computer search based on the application information retrieve and (8) parallel program generator.

A user develops an application software or employs a legacy software to solve his/her problem. The application software is first registered to the PSE server by the register component in the PSE toolkit. When it is registered, the PSE portal asks the user to input additional information for the application software. The input information includes the target problem, the solving area, restrictions, input parameters and their typical values, requirements, for example, library packages required, compiler requirement, memory and hard disk requirements and computer architecture requirement. The application and application register subcomponent in the PSE toolkit also supports making an application portal WEB page for the input data specification and for the explanation of the application. Then, the registered application is stored into the PSE application pool with the other information specified.

When the user runs the registered application, the user obtains the hardware and software information of the distributed computers, which should be supplied by an administrator of the distributed computer system. At the same time, preferable computers are suggested to the user, if they request. In the PSE server, the application software is first sent to the selected computers for trial compilation in order to check if the software can be compiled on the computers.

Next the user designs a WF for his/her whole problem solving. The WF search and register component has the case DB, in which user's newly developed WFs and legacy WFs are stored. The user can also select the legacy WF from the case DB and modify it for his/her problem.

Then, the application is deployed onto the target computers by the PSE deployment subcomponent. The WF designed by the user is executed to obtain results (Figure 3). The location of the results obtained is announced to the user so the user can retrieve, analyse or visualise the results.

3.4 Submodule of parallel program generation

The PSE toolkit includes a parallel program generation subcomponent, called NCAS (Boonemee and Kawata, 1998a,b; Kawata et al., 1994). The NCAS component helps users to generate MPI-based parallel simulation programs based on partial differential equations (PDEs). The NCAS capability explores possibilities to visualise and steer the parallel program design process. At present, NCAS supports a domain decomposition in a design of a parallel numerical simulation program and the domain decomposition is designed or steered by users through a visualisation window (Figure 4). After designing the

228 S. Kawata et al.

domain decomposition, the parallel program itself is designed and generated in NCAS and the designed parallel program is visualised and steered by a PAD diagram. In NCAS, MPI functions are employed for message passing and a single program multiple data (SPMD) model is supported. The visualisation and steering capabilities provide users a flexible design possibility of parallel programming.





Figure 4 MPI-based parallel program generation component in the PSE toolkit. Users can design and generate parallel programs for PDEs-based problems, and the processes were visualised and steered. Parallel programs generated are registered and deployed into the distributed computers through the PSE server



The parallel program generation module of NCAS inputs PDEs, boundary and initial conditions, parallel program information for a domain decomposition, CPU number employed and mesh information. All the processes are visualised and steered in order to meet the user's requirement. Figure 5 shows an example of the visualisation for a generated program by a PAD diagram. At the same time the program generation module also generates a document for the generated program as shown in Figure 6, because the NCAS component keeps the entire information from the abstract problem description, equations, etc., as described above. The generated program is registered in the PSE server and deployed into distributed computers. This program generation function in the PSE server may also help users work on parallel programming and on the distributed computing.

Figure 5 Example visualisation of a main part of a designed parallel program by using a PAD diagram in the parallel program generation component in the PSE toolkit



Figure 6 Example of a document generated by the parallel program generation component in the PSE server. This module contains the input information of abstract problem input information, for example, PDEs, problem area, variables, etc. Therefore, the document is also generated with the MPI-based parallel program from the higher level information



4 Example application processing flow and discussions

In this section, an example case is presented to demonstrate a viability of the PSE server. In this example, a two-dimensional heat transfer problem is solved in parallel (Figure 7). This example was performed on distributed computers located in a campus. First, a user logins the PSE server. A parallel program is designed and generated by the parallel program generation component of NCAS (Figure 5) and is registered as shown in Figures 2 and 8. Then, the PSE server presents candidate computers and select the computers among them. The application is successfully deployed and a WF is designed. The WF is run as shown in Figure 3. Final, the results shown in Figure 7 are obtained. Figure 7 shows an example temperature distribution and a speedup ration versus CPU number used and in this case homogeneous network-liked computers were used.

Figure 7 Example parallel computation results on the PSE server



Figure 8 One of application simulation program register windows in the PSE server

NAREGI/PSE Application Register				
Definition	Definition of Application			
Deploy	Application Name	APL01	KOKOKOESOKOKOKOESOKOKOKOE	
Complie	Directory	/aplsrv/aplpool/work/APL01		
TestRun	CompileComman	make – f Makefil		
Plug-in/Register	TestRunCommans	TestRun.sh		
		CK	Close	

In this specific example case, one application software generated by the parallel program generation subcomponent is employed to evaluate the PSE server. The present version of the PSE server does not support a job queue information retrieval, because we tested the PSE server on distributed computers in one campus and assumed that all the computers we used had no running jobs on the computers beforehand. In addition, we assumed that the target problem can be described by one computer simulation program and therefore the present PSE server does not support multiscale/multiphysics simulations which are consisted of multiple programs. In the near future these capabilities will be supported in the PSE server as well as other services described in this paper as GRID services (Global Grid Forum). At present the PSE server is implemented by JAVA and the parallel program generator is written in C++.

The PSE sever has been studied in part in a framework of National Research Grid Initiative (NaReGI). In NaReGI WF management system, super scheduler, middleware, networking, security, user management and nanoscience applications have been also studied intensively in parallel as well as the PSE server. Therefore, at present, we employ KSA/WI (KSA, 2002) for a WF engine.

5 Conclusion

In this paper, we presented the PSE server to help users to work on distributed computers. The role and viability of PSE in the distributed computer system are demonstrated. The PSE server provides a smooth and flexible environment in the HPC on the distributed computers and encapsulates the complex information of distributed system so that on the PSE server users can perform HPC as if distributed computers are under users' hand. The PSE server may open a new environment for HPC world.

Acknowledgement

The authors would like to express their appreciations to Prof. K. Miura (leader of NaReGI project), Dr K. Minami (Fujitsu Ltd.) and Dr T. Oroguchi (Fujitsu Ltd.) for their fruitful discussions on this subject. This work was supported in part by National Research GRID Initiative project (NaReGI).

References

- Boonmee, C. and Kawata, S. (1998a) 'Computer-assisted simulation environment for partial-differential-equation problem: 1. data structure and steering of problem-solving process', Transactions of Japan Society of Computer Engineering and Science, Paper No. 19980001. Available at: http://save.k.u-tokyo.ac.jp/jsces/trans/trans 1998/n19980001/ n19980001.html.
- Boonmee, C. and Kawata, S. (1998b) 'Computer-assisted simulation environment for partial-differential-equation problem: 2. Visualisation and steering of problem-solving process', Transactions of Japan Society of Computer Engineering and Science, Paper No. 19980002. Available at: http://save.k.u.tokyo.ac.jp/jsces/trans/trans1998/n19980002/ n19980002.html.

230 S. Kawata et al.

CACTUS. Available at: http://www.cactuscode.org.

Gallopoulos, E., Houstis, E.N. and Rice, J.R. (1994) 'Computer as thinker/doer: problem-solving environments for computational science', *IEEE Computer Science and Engineering*, Vol. 1, pp.11–23.

Global Grid Forum. Available at: http://www.gridforum.org.

- Globus Toolkit. Available at: http://www.unix.globus.org/toolkit.
- Houstis, E.N., Gallopoulos, E., Rice, J.R. and Bramley, R. (Eds) (2000) *Enabling Technologies for Computational Science*, Amsterdam: Kluwer Academic Publishers.
- Kadooka, Y., Kobashi, H., Choi, J.W., Lee, Y.H. and Tago, Y. (2003) 'PIV web visualisation system toward PIV visualisation grid', *Journal of Visualisation*, Vol. 6, No. 3, pp.283–291.
- Kawata, S., Iijima, K., Boonmmee, C. and Manabe, Y. (1994) 'Computer-assisted scientific-computation/simulationsoftware development system—including a visualisation system', *IFIP Transaction*, Vol. A48, pp.145–153.
- Konno, C., et al. (1987) 'Automatic code generation method of DEQSOL', Journal of Information Processing, Information Processing Society of Japan, Vol. 11, No. 1, pp.15–21.
- KSA/WI. Available at: http://pr.fujitsu.com/jp/news/2002/ 09/181. html.
- Message Passing Interface Forum. Available at: http://www.unix. mcs.anl.gov/mpi/mpich.

- National Research Grid Initiative (NaReGI). Available at: http://www.naregi.org/index_e.html.
- Schuchardt, K., et al. (2002) 'Ecce; a problem-solving environments evolution toward grid services and a web architecture'. Available at: http://ecce.emsl.pnl.gov/docs/pub lications/ecce_evolution.pdf.
- Teramoto, T., Nakamura, T., Kawata, S., Matide, S., Hayasaka K., Nonaka, V., Sasaki, E. and Sanada, Y. (2001) 'A distributed problem-solving environment (PSE) for partial differential equation based problems', Transactions of Japan Society of Computer Science Engineering, Paper No. 20010018. Available at: http://save.k.u-tokyo.ac.jp/ jsces/trans/trans2001/No20010018.pdf.
- Umetani, Y., Konno, C., Ohta, T. (1991) 'Visual PDEQSOL: a visual and interactive environments for numerical simulation', Proceedings of IFIP TC2/WG2.5 Working Conference on Programming Environments for High-Level Scientific Problem-solving, Karlsruhe, Germany, pp.259–267.
- Umetani, Y., Tsuji, M., Iwasawa, K. and Hirayama, H. (1987) 'DEQSOL: a numerical simulation language for vector/ parallel processors', *Proceedings of IFIP WG2.5 Working Conference on Problem-solving Environments for Scientific Computing*, Sophia Antipolis, France, pp.147–162.

Unicore Forum. Available at: http://www.unicore.org/.