Cooperative fault-tolerant distributed computing

U.S. Department of Energy Grant DE-FG02-02ER25537
Final Report

Vaidy Sunderam
Department of Mathematics and Computer Science
Emory University, Atlanta, GA 30322, USA
vss@mathcs.emory.edu

1 Overview

The primary goals of the Harness project were to devise effective solutions to two of the most pressing issues in large scale high performance computing: aggregating collections of widely distributed resources and dealing with failures that are inevitable in such environments. Our work aims to realize the first objective through cooperative resource sharing software that spans multiple administrative and geographic domains, emphasizing secure and controlled resource management while being resilient to failures at the infrastructure level. To complement this software infrastructure, a programming model and framework that offers a fault-tolerant enhancement to the standard message passing model is proposed. Our efforts are a natural continuation of prior metacomputing research at Emory University, Oak Ridge National Laboratory, and the University of Tennessee, and is a follow-on to the PVM system for heterogeneous computing, and subsequently, the Harness system that investigated dynamically reconfigurable software frameworks. The three institutions involved in this project have worked closely together; our project methodology has been driven by regular meetings of all personnel, where technical design, operational modes, compatibility aspects, and integration issues are discussed in depth, for completed and ongoing work as well as for planned efforts at each individual site. The subprojects undertaken at each location are closely linked and address complementary aspects of the project; this report describes work done during the project at Emory University and summarizes our accomplishments.

2 Overall Progress

The Emory focus within the overall project has been on distributed computing software substrates that permit flexible but robust aggregation of resources from multiple domains, and at the same time, support customizable parallel programming models. During the project, we designed the metacomputing architecture for such a framework, drawing upon a component-oriented approach that was successfully demonstrated as viable in the previous Harness I project. We also developed implementations of some of the low level software substrate and communication mechanisms. In the later stages of the project, we concentrated on (1) scalable and stateless provider-centric designs for resource sharing, with a view to large-scale aggregation with resilience to failures; (2) attaining high efficiency in terms of minimizing CPU overheads due to the system software, and delivering high communication performance through protocol dynamic selection; and (3) designs for integrating the distributed computing software layer with programming environments, including a service-oriented paradigm and the FT-MPI model.

2.1 Provider-centric resource sharing

Recent experiences with large scale distributed systems and grids have highlighted the complexity and cumbersome nature of monolithic software systems for resource sharing. Our challenge was therefore to evolve a distributed computing architecture that is simultaneously lightweight, scalable, and resilient to failures – but yet permits controlled resource sharing across multiple administrative domains. The result of our efforts in this respect is the H2O substrate that forms the low level underpinning of the Harness II environment. In the H2O system, a software backplane architecture supporting component-based services hosts pluggable components that provide composable services. Such components may be uploaded by resource providers but also by clients or third-parties who have appropriate permissions. This aspect allows for virtualizing resources in a shape and form that is most suitable to the targeted application. It also permits the deployment of different parallel programming platforms or runtime support systems that may be needed – in fact, this facility is exploited when running FT-MPI and MPICH programs under Harness.

In terms of sharing, resource owners retain complete and fine-grained control over policies; yet, authorized clients have substantial flexibility in configuring and securely using compute, data, and application services. By utilizing a model in which service providers are central and independent entities, global distributed state is significantly reduced at the lower levels of the system, thereby resulting in increased failure resilience and dynamic adaptability. Further, since providers themselves can be clients, a distributed computing environment can be created that operates in true peer-to-peer mode (without the need for centralized control or administration), but offers an arbitrary network of resource sharing relationships.

Harness II essentially constructs a concurrent computing layer on a collection of H2O "kernels", which, in the interest of statelessness and scalability, do not maintain any state concerning aggregation. Harness II uses the well-established Distributed Virtual Machine model that was first introduced in PVM but does so by deploying a collection of pluglets (pluggable components) on a client-selected set of H2O kernels. Thus, in Harness II, DVM's may include resources belonging to independent administrative domains and provided by collaborating contributors. This of course raises important issues of trust management and appropriate security mechanisms. However, once again, by isolating resource sharing relationships to one client and one provider at a time, such arrangements are greatly simplified and also suitable to the situation at hand. For example, H2O provides simple sub-domain based authentication and authorization for sharing clusters between different departments within one university, but also supports certificate based authentication and detailed policy management for sharing resources in less informal settings.

2.2 Efficient Resource Delivery and Communications Performance

The H2O architecture is service-oriented; its fundamental abstractions are the *kernel*, being the service container, and the *pluglet*, being a service instantiated within the container. H2O enables resource owners to define access policies for using and deploying services on their kernels. For high-performance computing, a common resource type is raw CPU access. (Other raw resources such as data stores, as well as "value-added" resources in the form of libraries, solvers, or entire applications may also be wrapped within a pluglet abstraction and offered as a service). The major research issues that we addressed in this regard concerned: (1) delivery of the underlying resource represented by the pluglet service abstraction to the user of the resource with as little overhead as possible; (2) while simultaneously adhering to the resource provider's sharing policy and ensuring protection. We approached this task by defining a set of provider policy specifications, corresponding to authorization to access provider resources on a per-client basis. Some policy targets are specified in boolean terms, while others (such as storage) are quantified, and a few are enumerated. Due to the high degree of complexity of implementing fine-grained access to OS scheduled resources (e.g. CPU), we decided to implement only temporal policies for such parameters, augmented by higher level controls. Thus, to dedicate a CPU resource to a certain client, the policy statement would issue a complementary edict, i.e. to prohibit all other clients during that period. We determined through a number of experiments and analyses that attempting to exercise more control over such resources leads to unnecessarily large overheads and perturbation, and in any case, is rarely useful for high performance applications. We have thus implemented an effective policy and resource management mechanism in H2O that gives providers strong control over their resources, but imposes little overhead when clients actually use them. In experiments with pluglet interfaces to various services, between zero and six percent overhead was observed.

The second issue relating to efficiency that we addressed was one that occurs in all

service-based environments. Since clients and providers have no a-priori knowledge of each other's interaction capabilities, it is often necessary to utilize the most generic communication protocol for service access – but this is almost always very inefficient. Our solution to this situation is embedded in the RMIX framework for access to pluglet services. This framework offers the well established and well-understood RMI paradigm for access to remote services, but augments these basic facilities with significant extensions suitable for use in high performance distributed computing. During the second project period, we completed the detailed design of this layer, and implemented efficient protocol switching. Using this technique, a service is initially accessed using (X)SOAP as the first contact protocol, but immediately followed by negotiation to dynamically switch to the most efficient protocol supported by both ends of the communication. Furthermore, these transports are themselves pluggable, thereby enabling new modules to be developed and deployed, as appropriate to a given circumstance. We have implemented JRMP and RPCX (extended RPC) as example transports; using the latter, data transfer speeds within ten percent of that attainable for strongly-typed data across heterogeneous platforms have been demonstrated. We have also begun to design pluggable transport modules for high-speed cluster networks such as GigE and Myrinet to support efficient pluglet interaction in such environments. It should be noted, however, that pluglets implementing parallel programming environments (e.g. FT-MPI or MPICH) may continue to use their native communication mechanisms and may bypass H2O communication channels if so desired. In the context of RMIX, we have also developed a semantically strong asynchronous invocation model. This feature enables traditional message passing applications to map naturally and in a one-to-one manner to a service-oriented infrastructure, and avail of several benefits without incurring the performance penalty of the request-response paradigm.

2.3 Integration with Programming Environments

Like its predecessor Harness I, the Harness II/H2O framework is a component-based infrastructure where different programming models may be deployed (and co-exist) on a common runtime substrate. In H2O, a concurrent environment (e.g. PVM, MPICH, DSM etc) may be constructed by writing appropriate pluglets that emulate or provide runtime facilities needed by those platforms. However, pluglets may themselves be used to program applications following a style similar to that of RMI, and may also be used to offer standardized services, such as those based on OGSA. Both these models were designed and implemented and have been published in both tutorial and research paper form, for the benefit of users wishing to use these paradigms. To support OGSA compliance, the H2O system was augmented with a set of specially designed pluglets. Given a remotely accessible service written in pluglet-form, this auxiliary system, in conjunction with existing tools, generates WSDSL/GSDL descriptions of pluglet services, publishes these descriptions, and provides facilities for client lookup followed by handle generation, reference creation, and eventually, access to service methods.

Supporting traditional high-performance message-passing distributed computing has always been one of the main goals of this project. In the past project period, we focused considerable effort on coupling the underlying resource sharing and resource management infrastructure with MPI-based programming environments. In particular, an important objective is to support the FT-MPI model, whose API and runtime libraries are being developed by our partners at UT. Two approaches to integrate FT-MPI with H2O were explored. One involves loading FT-MPI daemons (name service, notifier, startup daemons, and watchdog) into the H2O kernel process space via JNI. The other approach is to use H2O as a startup and lifetime management subsystem for FT-MPI. H2O pluglets would launch FT-MPI daemons as separate processes (rather than loading shared objects into the H2O address space). FT-MPI daemons would continue to be responsible for communication and some aspects of security. This approach is attractive because it reduces duplication of function while leveraging authentication and authorization facilities in H2O. Using wrappers to cast FT-MPI name service and startup daemons as pluglets, this scheme was prototyped and tested successfully, demonstrating that the two subsystems could indeed be naturally combined into a framework that supports fault tolerant MPI on resources aggregated across multiple domains. In an independent subproject, we also implemented a H2O pluglet suite that enhances the standard MPICH2 distribution (a beta version was used) to permit operation across firewalls and non-routable networks. When stable, we will offer this solution as an alternative to other approaches for executing MPICH2 programs in such environments. These two efforts also successfully demonstrated the Harness II model of constructing distributed virtual machines (DVMs) by layering special pluglets on a collection of independent H2O kernels (H2O is a short acronym for Harness II operating environment). We believe that these exercises also demonstrate that such a clear separation between the resource sharing aspect on the one hand, and the concurrent programming environment on the other hand, is the key to creating scalable, failure resilient distributed computing platforms.

3 Research Output

Our work on the Harness II system has resulted in several novel contributions to distributed computing that we believe propose effective solutions to long standing problems. One of these is the H2O pluglet model that offers a user-customizable but secure and controlled way to share resources across multiple domains. Another is the RMIX approach to communication, that uses dynamic selection and protocol switching to avoid tradeoffs between generality and performance. These and other research artifacts are described in a number of refereed papers and presentations (listed below). In addition, the software itself has also been released to the community for potential use as a distributed high-performance computing platform.

3.1 Talks and Professional Activities

1. "Resource and Application Adaptivity in Message Passing Systems", invited talk, *13th EuroPVM/MPI Conference*, Bonn, Germany, September 2006.

2. "New Approaches to Distributed Middleware for Resource Sharing", invited talk, *6th International Workshop on Innovative Internet Community Systems (I2CS 2006)*, Neuchatel, Switzerland, June 2006.

3. "Metacomputing Revisited: Alternative Paradigms for Distributed Resource Sharing", invited keynote address, *International Conference on Computational Science (ICCS) 2006*, Reading, England, May 2006.

4. "Integrating heterogeneous information services using JNDI", contributed talk, *International Parallel and Distributed Processing Symposium (IPDPS-HCW) 2006*, Rhodes, Greece, April 2006.

5. "Virtualization Issues in MetaSystems", invited lecture, *University of Genoa Seminar Series*, Genoa, Italy, March 2006.

6. "Virtualization Issues in MetaSystems", invited keynote lecture, *2005 Cracow Grid Workshop*, Krakow, Poland, November 2005.

7. "Alternative Approaches to Metacomputing: Harness and H2O", invited keynote lecture, *2005 High Performance Computing and Communications Conference (HPCC 2005)*, Sorrento, Italy, September 2005.

8. "Virtualization in Parallel Distributed Computing", invited keynote lecture, *12th EuroPVM/MPI Conference*, Sorrento, Italy, September 2005.

9. "Performance and Client Heterogeneity in Service-based Metacomputing", contributed paper, *International Parallel and Distributed Processing Symposium (IPDPS-HCW) 2004*, Santa Fe, NM, April 2004.

10. "Semantic Aspects of Asynchronous RMI: the RMIX Approach", contributed paper, *International Parallel and Distributed Processing Symposium (IPDPS-JPDC) 2004*, Santa Fe, NM, April 2004.

11. "Efficient Monitoring to Detect Channel Failures for MPI Programs", contributed paper, *Proceedings of the 12th Euromicro Conference on Parallel, Distributed and Network based Processing*, A Coruna, Spain, pp. 85-92, February 2004.

12. "Programming Environments for Grids and Metacomputing Systems", invited tutorial, *10th EuroPVM/MPI Conference*, October 2003.

13. "Parallel Programming in Grids and Metacomputing Systems", invited talk, *Fifth International Conference on Parallel Processing and Applied Mathematics*, Czestochowa, Poland, September 2003.

14. "Current Trends in Parallel Distributed High Performance Computing", invited keynote lecture, *Workshop on Parallel and Distributed Scientific and Engineering Computing and Applications 2003*, Nice, France, April 2003.

3.2 Publications

1. D. Kurzyniec, V. Sunderam, "Failure Resilient Heterogeneous Computing Across Multidomain Clusters", International Journal of High Performance Computing Applications, Vol. 19, No. 2, pp. 143-156, Summer 2005.

2. Magdalena Slawinska, Dawid Kurzyniec, Jaroslaw Slawinski, Vaidy Sunderam, "Zero-Force MPI: Towards Tractable Toolkits for High Performance Computing", *Poster presentation, Supercomputing 2006 (SC06)*, Tampa, FL, November 2006.

3. D. Dewolfs, J. Broeckhove, V. Sunderam, and G.E. Fagg, "FT-MPI, Fault-Tolerant Metacomputing and Generic Name Services: A Case Study", *Proceedings 2006 EuroPVM/MPI Conference*, Bonn, Germany, pp. 133-140, September 2006.

4. M. Malawski, M. Bubak, M. Placek, D. Kurzyniec, V. Sunderam "Experiments with Distributed Component Computing Across Grid Boundaries", *Proceedings 15th IEEE Intl. Symposium on High Performance Distributed Computing – HPDC 2006 (HPC-GECO Workshop)*, Paris, France, June 2006.

5. Dirk Gorissen, Piotr Wendykier, Dawid Kurzyniec, Vaidy Sunderam, "Integrating heterogeneous information services using JNDI", *International Parallel and Distributed Processing Symposium (IPDPS-HCW) 2006*,Rhodes, Greece, April 2006.

6. P. Jurczyk, M. Golenia, M. Malawski, D. Kurzyniec, Marian Bubak, V. Sunderam, "Enabling Remote Method Invocations in Peer-to-Peer Environments: RMIX over JXTA", *Sixth International Conference on Parallel Processing and Applied Mathematics (PPAM 2005)*, Poznan, Poland, September 2005.

7. D. Dewolfs, D. Kurzyniec, V. Sunderam, J. Broeckhove, T. Dhaene, and G.E. Fagg, "Applicability of Generic Naming Services and Fault-Tolerant Metacomputing With FT-MPI", *Proceedings 2005 EuroPVM/MPI Conference*, Sorrento, Italy, September 2005.

8. Maciej Malawski, Dawid Kurzyniec, Vaidy Sunderam, "MOCCA – Towards a Distributed CCA Framework for Metacomputing", *International Parallel and Distributed Processing Symposium (IPDPS-HIPS) 2005*, Denver, CO, April 2005.

9. G. Stuer, J. Broeckhove, V. Sunderam, "Towards OGSA Compatibility in Alternative Metacomputing Environments", Journal of Future Generation Computer Systems, Vol. 21, No. 1, pp. 221-226, January 2005.

10. Dawid Kurzyniec, Vaidy Sunderam, "Combining FT-MPI with H2O: Fault-Tolerant MPI Across Administrative Boundaries", *International Parallel and Distributed Processing Symposium (IPDPS-HCW) 2005*, Denver, CO, April 2005.

11. Peter Hwang, Dawid Kurzyniec, Vaidy Sunderam, "Heterogeneous Parallel Computing Across Multidomain Clusters", *Proceedings 2004 EuroPVM/MPI Conference*, Budapest, Hungary, September 2004.

12. Tomasz Wrzosek, Dawid Kurzyniec, and Vaidy Sunderam, "Performance and Client Heterogeneity in Service-based Metacomputing", *International Parallel and Distributed Processing Symposium (IPDPS-HCW) 2004*, Santa Fe, NM, April 2004.

13. Dawid Kurzyniec and Vaidy Sunderam "Semantic Aspects of Asynchronous RMI: the RMIX Approach", *International Parallel and Distributed Processing Symposium (IPDPS-JPDC) 2004*, Santa Fe, NM, April 2004.

14. Z. Nemeth, V. Sunderam, "Characterizing Grids: Attributes and Formalisms", *Journal of Grid Computing*, Vol. 1, No. 1, pp. 9-23, 2003.

15. G. Stuer, J. Broeckhove, V. Sunderam, "Publishing H2O Pluglets in UDDI Registries", Journal of Computing and Informatics, Vol. 23, No. 4, 2004.

16. Tomasz Ampula, Dawid Kurzyniec, Vaidy Sunderam, Henryk Witek, "The Genetic Algorithms Population Pluglet for the H2O Metacomputing System", *International Conference on Computational Science (ICCS 2004)*, Krakow, Poland, June 2004.

17. Tomasz Wrosek, Dawid Kurzyniec, and Vaidy Sunderam, "Performance and Client Heterogeneity in Service-based Metacomputing", *International Parallel and Distributed Processing Symposium (IPDPS-HCW) 2004*, Santa Fe, NM, April 2004.

18. Dawid Kurzyniec and Vaidy Sunderam "Semantic Aspects of Asynchronous RMI: the RMIX Approach", *International Parallel and Distributed Processing Symposium (IPDPS-JPDC) 2004*, Santa Fe, NM, April 2004.

19. Elsa Macias, Alvaro Suarez, and Vaidy Sunderam, "Efficient Monitoring to Detect Channel Failures for MPI Programs", *Proceedings of the 12th Euromicro Conference on Parallel, Distributed and Network based Processing*, A Coruna, Spain, pp. 85-92, February 2004.

20. V. Sunderam, D. Kurzyniec, T. Wrzosek, "Towards Self-Organizing Distributed Computing Frameworks", *Journal of Parallel Processing Letters*, Vol. 13, No. 2, June 2003.

21. Gunther Stuer, Vaidy Sunderam, Jan Broeckhove, "Towards OGSA Compatibility in Alternative Metacomputing Frameworks", *International Conference on Computational Science (ICCS 2004)*, Krakow, Poland, June 2004.

22. Tomasz Wrzosek, Dawid Kurzyniec, Dominik Drzewiecki, and Vaidy Sunderam, "Resource Monitoring and Management in Metacomputing Environments", *Proceedings 10th European PVM/MPI User's Group Meeting*, Venice, Italy, pp. 629-635, September 2003.

23. Vaidy Sunderam, James Pascoe and Roger Loader, "Towards a Framework for Collaborative Peer Groups", *Third IEEE International Symposium on Cluster Computing and the Grid (CCGRID-GP2PC) 2003*, pp. 428-433, May 2003.

4 Summary

This report has outlined the Harness II project accomplishments at Emory University. Additional information or copies of publications may be obtained from the principal investigator. U.S. Department of Energy support of our research efforts is greatly appreciated.