A Synopsis of the Legion Project

Andrew S. Grimshaw
William A. Wulf
James C. French
Alfred C. Weaver
Paul F. Reynolds Jr.

Technical Report No. CS-94-20
June, 1994

A Synopsis of the Legion Project

e pluribus unum -- one out of many

Andrew S. Grimshaw, William A. Wulf, James C. French, Alfred C. Weaver, Paul F. Reynolds Jr.
Department of Computer Science, University of Virginia

Abstract

The coming of giga-bit networks makes possible the realization of a single nationwide virtual computer comprised of a variety of geographically distributed high-performance machines and workstations. To realize the potential that the physical infrastructure provides, software must be developed that is easy to use, supports large degrees of parallelism in applications code, and manages the complexity of the underlying physical system for the user. This short paper briefly describes our approach to constructing and exploiting such "metasystems". Our approach inherits features of earlier work on parallel processing systems and heterogeneous distributed computing systems. In particular, we are building on Mentat, an object-oriented parallel processing system developed at the University of Virginia. A more detailed presentation can be found in technical report CS 94-21, "Legion: The Next Logical Step Towards a Nationwide Virtual Computer".

1.0 Introduction

The information superhighway is upon us – it will provide the communication infrastructure for applications as yet undreamed. But what will the highway connect? Will it connect separate islands of computational service, allowing them to no more than exchange information? Or will it allow integration of a multitude of islands into a single monolithic virtual machine? Our goal is to create a single nationwide metasystem called Legion by combining the communications infrastructure of the NII with the computational and data resources already available.

Legion will consist of workstations, vector supercomputers, and parallel supercomputers connected by local area networks, enterprise-wide networks, and the National Information Infrastructure. The total computation power of such an assembly of machines is enormous, approaching a petaflop; this massive potential is, as yet, unrealized. These machines are currently tied together in a loose confederation of shared communication resources used primarily to support electronic mail, file transfer, and remote login. However, these resources could be used to provide far more than just communication services; they have the potential to provide a single, seamless, computational environment in which processor cycles, communication, and data are all shared, and in which the workstation across the continent is no less a resource than the one down the hall.

A Legion user has the illusion of a single, very powerful computer[1] on her desk. It is Legion's responsibility to *transparently* schedule application components on processors, manage data transfer and coercion, and provide communication and synchronization in such a manner as to minimize[2] execution time via parallel execution of the application components. System bound-

1. We use computer in its broadest sense, to include any display or I/O device, including virtual reality interfaces such as head-mounted displays and data gloves.
2. In general this is NP-hard. We really mean "do a good job" using a heuristic.

aries will be invisible, as will the location of data and the existence of faults.

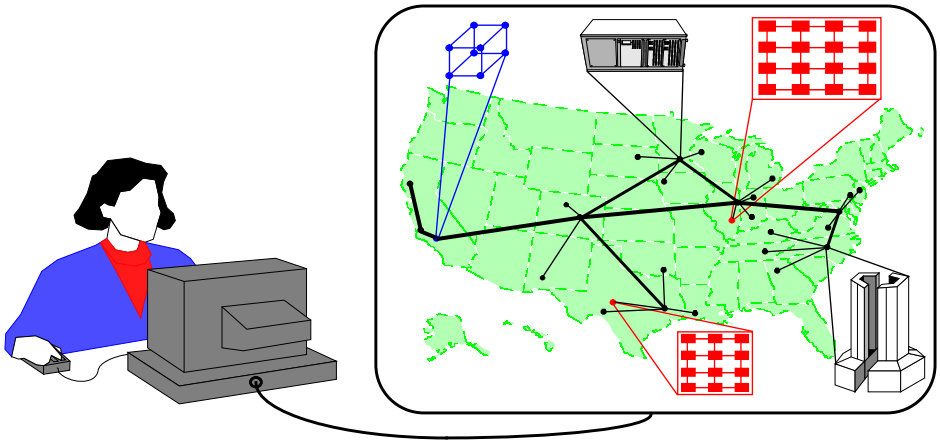

Figure 1 The user views Legion as a single computational resource.

The potential benefits of Legion are enormous. The benefits we envision include: (1) more effective collaboration by putting coworkers in the same virtual workplace; (2) higher application performance due to parallel execution and exploitation of off-site resources; (3) improved access to data and computational resources for smaller sites; (4) improved researcher and user productivity resulting from more effective collaboration and better application performance; (5) increased resource utilization; and (6) a considerably simpler programming environment for the applications programmers. Indeed, it seems probable to us that the NII can reach its full potential only with a Legion-like infrastructure.

Before the Legion vision can be realized, several technical challenges must be overcome. These are software problems; the hardware challenges are being addressed and are the enabling technologies that provide the opportunity. The software challenges revolve around eight central themes: *achieving high performance via parallelism, managing and exploiting component heterogeneity, resource management, file and data access, fault-tolerance, ease-of-use and user interfaces, protection and authentication, and exploitation of high-performance communications protocols*. We realize that these are serious, non-trivial, issues; we examine them in more detail in [19][22].

In addition to the purely technical issues, there are also political, sociological, and economic ones. These include encouraging the participation of resource-rich centers and the avoidance of the human tendency to free-ride. We intend to discourage such practices by developing and employing accounting policies that encourage good community behavior.

The vision of a seamless metasystem or metacomputer such as Legion is not novel. Indeed, a number of systems have been designed to attack one or more of the problems mentioned above, e.g., Andrew, Locus and NSF for file systems [32][35][44], Locus for fault-tolerance, Sun XDR and the University of Washington HCS for heterogeneity[39][40][42]. None has been fully successful. What has changed that makes the realization of a complete high performance metacomputer possible? The change is that achieving high performance via parallelism, previously available only for tightly coupled parallel processors, is now possible for loosely coupled distributed systems [1][3][4][6][8][14][27][33][37][45].

Whether or not a metasystem is explicitly constructed by design, the nation (and perhaps the world) will eventually build a system that shares at least some of the attributes of Legion. The reason is simple: individual and organizational users will be required to deal with the increasingly

obvious shortcomings of a computing infrastructure consisting of islands of computational power connected via the Internet. Internet tools such as *gopher, worldwide web* and *Mosaic* are examples of current attempts to bridge the gaps between local systems.

The issue is not whether metasystems will be developed; clearly they will. Rather, the question is whether they will come about by design and in a coherent, seamless system – or painfully and in an ad hoc manner by patching together congeries of independently developed systems, each with different objectives, design philosophies, and computation models.

2.0 Legion

From our vision of Legion we have distilled six primary design objectives that are central to the success of the project; easy-to-use, seamless computational environment; high performance via parallelism; single, persistent namespace; security for both users and resource providers; manage and exploit resource heterogeneity; and minimal impact on resource owner's local computation.

Easy-to-use, seamless computational environment. Legion must mask the complexity of the hardware environment and the complexity of communication and synchronization of parallel processing. Machine boundaries should be invisible to users. As much as possible, compilers, acting in concert with run-time facilities, must manage the environment for the user.

High performance via parallelism. Legion must support easy-to-use parallel processing with large degrees of parallelism. This includes task and data parallelism and their combinations. Because of the nature of the interconnection network, Legion must be latency tolerant. Further, Legion must be capable of managing hundreds or thousands of processors.

Single, persistent namespace. One of the most significant obstacles to wide area parallel processing is the lack of a single name space for file and data access. The existing multitude of disjoint name spaces makes writing applications that span sites extremely difficult.

Security for users and resource owners. Because we cannot replace existing host operating systems, we cannot significantly strengthen existing operating system protection and security mechanisms. However, we must ensure that existing mechanisms are not weakened by Legion.

Manage and exploit resource heterogeneity. Clearly Legion must support interoperability between heterogeneous components. In addition, Legion will be able to exploit diverse hardware and data resources. Some architectures are better than others at executing particular kinds of code, e.g., vectorizable codes. These affinities, and the costs of exploiting them, must be factored into scheduling decisions and policies.

Minimal impact on resource owner's local computation. The *noticeable* impact of Legion on local resources must be small, particularly with regard to interactive sessions. If users notice a significant performance penalty when their site is attached to Legion, they will withdraw; an observed penalty must be more than offset by the benefits of Legionnaire status.

2.1 Approach

The principles of the object-oriented paradigm are the foundation for the construction of Legion; our goal will be exploitation of the paradigm's encapsulation and inheritance properties, as well as benefits such as software reuse, fault containment, and reduction in complexity. The need for the paradigm is particularly acute in a system as large and complex as Legion. Other investigators have proposed constructing application libraries and applications for wide-area parallel processing using only low-level message passing services. Use of such tools requires the programmer to address the full complexity of the environment; the difficult problems of managing faults, scheduling, load balancing, etc., are likely to overwhelm all but the best programmers.

Our approach to constructing Legion is evolutionary rather than revolutionary. We have begun by first constructing a Legion testbed by extending Mentat, an existing object-oriented parallel processing system [24]. Mentat attacks the problem of providing easy-to-use high performance parallelism to users. Mentat has been used to implement several real-world applications on hardware platforms spanning the bandwidth/latency space and in a heterogeneous environment [20][21][23]. Mentat's object-oriented structure, and its ability to achieve high-performance on platforms with very different communications characteristics are the key factors in our choice of Mentat as our implementation vehicle. The testbed provides us with an ideal platform to rapidly prototype ideas, forcing the details and hidden assumptions to be carefully examined, and exposing flaws in the ideas or in the system components.

There are two principal reasons for extending an existing system rather than starting work on Legion from scratch. First, building on Mentat will allow very substantial savings in the amount of code required before initial applications can be executed. Second, we will be able to use a system which we know works and with which we already have had very considerable experience. New capabilities can be added as new problems are addressed and their solutions incorporated.

Finally, our model for the evolution of Legion is that of the Internet. We will begin with a campus-wide virtual computer here on our own campus, then expand to a small community of participating sites. Legion will be an open system, rather than an exclusive club.

3.0 Agenda

Our agenda consists of three stages: (1) the construction of a campus-wide virtual computer at the University of Virginia, (2) packaging the campus-wide system for preliminary experimentation and use by Legionnaires, and (3) expansion to a nationwide demonstration system. Each of these three stages will build upon the previous.

Before any major project is undertaken, one must ask how to measure success. In parallel processing, success is measured by application performance (speedup, MFLOPS) and the flexibility and ease of use of the tool. These are important metrics for Legion as well, but they are not the only metrics. Other important metrics include acceptance by the user community, fault-tolerance, cost per *used* MIP/FLOP, and whether tasks can be performed that were not possible before.

Application performance will be measured for a variety of real-world applications, as well as selected kernel codes and parallel processing benchmarks. The applications will be drawn from a diverse set of disciplines: biology, physics, electrical engineering, chemistry, economics, radio astronomy, and command and control. The applications will possess different granularity characteristics, as well as different latency tolerances. It is not our intent, however, to show that all applications will be capable of exploiting the nationwide resources of Legion. Some applications, those with inherently small granularity or that are latency intolerant, will remain best suited to local operation, e.g., on a single processor or on a single tightly-coupled parallel processor.

3.1 Construction of a Campus-Wide Virtual Computer (CWVC)

The campus-wide virtual computer is a direct extension of Mentat to a larger scale, and is a prototype for the nationwide system. Even though the CWVC is much smaller, and the components much closer together, than in the envisioned nationwide Legion, it still presents many of the same challenges. The processors are heterogeneous, the interconnection network is irregular, with orders of magnitude differences in bandwidth and latency, and the machines are currently in use for on-site applications that must not be negatively impacted. Each department operates essentially as an island of service, with its own NFS mount structure.

Table 1-a. Resources

Computer Type	Quantity
SPARC IPC	38
SPARC 1+	1
SPARC2	13
SPARC10	2
SGI Indigo	6

Table 1-b. Complib - 42,864 sequence target library

Number of workers	Best Time (sec)	Average Time (sec)	Best Total Time (sec)	Average Total Time (sec)
sequential (IPC)			10,876	
10	567	733	595	769
20	841	874	892	927
30	636	759	693	828
40	441	467	544	546
50	398	443	481	509
60	343	411	443	450
70	323	332	376	387

The CWVC is both a prototype and a demonstration project. The objectives are to demonstrate the usefulness of network-based, heterogeneous, parallel processing to university computational science problems; provide a shared high-performance resource for university researchers; provide a given level of service (as measured by turn-around time) at reduced cost; and act as a testbed for the nationwide Legion.

The prototype consists of over sixty workstations and is now operational. In [18] we present the performance of two production applications that we have used to test the efficacy of our approach: *complib*, a biochemistry application that compares DNA and protein sequences, and ATPG, an electrical engineering application that generates test patterns for VLSI circuits. The performance results are encouraging. Table 1 presents early performance results for *complib* taken from [18].

3.2 The Nationwide Demonstration Project

Now that the CWVC has successfully demonstrated the efficacy of our approach on a small scale, we will turn our attention to the nationwide Legion. The first step is to identify potential member organizations that would be interested in participating in the demonstration project. Researchers at NASA JPL, Sandia (at Livermore), Los Alamos, Oregon State, Indiana University, and Emory have agreed to participate. Now that a sufficient number of sites have signed on, the next step is to attend to the interconnection network. To be successful on any but the most trivially parallel applications, a high-bandwidth network between the sites and high-performance protocols for the network are necessary. This is a key aspect and is already under investigation.

4.0 Conclusion

Legion is an ambitious project. If the CWVC is successful it will significantly increase the computational resources available to university researchers, increase throughput, simplify cross-disciplinary collaboration within the university, and increase the productivity of local research. The nationwide Legion, if successful, will permit the nation to fully realize the potential offered by the NII and the tremendous aggregate data and computing resources available. Legion has the potential to usher in a new era of computing. Our approach builds on advances in both parallel and distributed computing, and on our earlier work in object-oriented parallel processing. A prototype has been constructed at the University of Virginia, and several Legion sites have agreed to participate in a nationwide system.

5.0 References

[1] S. Ahuja, N. Carriero, and D. Gelernter, "Linda and Friends," *IEEE Computer*, pp. 26-34, August, 1986.

[2] H. Bal, J. Steiner, and A. Tanenbaum, "Programming Languages for Distributed Computing Systems," *ACM Computing Surveys*, pp. 261-322, vol. 21, no. 3, Sept. 1989.

[3] J. Boyle et al., *Portable Programs for Parallel Processors*, Holt, Rinehart and Winston, New York, 1987.

[4] A. Beguelin et al., "A Users' Guide to PVM (Parallel Virtual Machine)", Oak Ridge National Laboratory TM-11826.

[5] B. N. Bershad, et al., "A Remote Procedure Call Facility for Interconnecting Heterogeneous Computer Systems," *IEEE Trans. Software. Eng. SE*, vol. 13, no. 8, pp. 880-894, August, 1987.

[6] F. Bodin, et. al., "Distributed pC++: Basic Ideas for an Object Parallel Language," *Proceedings Object-Oriented Numerics Conference*, pp. 1-24,Sunriver, Oregon, April 25-27, 1993.

[7] J. C. Browne, T. Lee, and J. Werth, "Experimental Evaluation of a Reusability-Oriented Parallel Programming Environment," IEEE Transactions on Software Engineering, pp. 111-120, vol. 16, no. 2, Feb., 1990.

[8] D. Callahan and K. Kennedy, "Compiling Programs for Distributed-Memory Multiprocessors" *The Journal of Supercomputing*, no. 2, pp. 151-169, 1988, Kluwer Academic Publishers.

[9] N. Carriero and D. Gelernter, "Linda in Context," *Communications of the ACM*, vol. 32, no. 4, pp. 444-458, April, 1989.

[10] N. Carriero, and D. Gelernter, "How to Write Parallel Programs: A Guide to the Perplexed," *ACM Computing Surveys*, pp. 91-125, vol. 23, num. 1, March. 1991.

[11] N. Carriero, D. Gelernter, and T.G. Mattson, "Linda in Heterogeneous Computing Environments," *Proceedings of WHP 92 Workshop on Heterogeneous Processing*, IEEE Press, pp. 43-46, Beverly Hills, CA, March, 1992.

[12] M.J. Carey, et. al., "Shoring Up Persistent Applications," *to appear, SIGMOD 1994*.

[13] T. L. Casavant, and J. G. Kuhl, 'A Taxonomy of Scheduling in General-Purpose Distributed Computing Systems,' *IEEE Transactions on Software Engineering*, vol. 14, pp. 141-154, February, 1988.

[14] A.L.Cheung, and A.P. Reeves, "High Performance Computing on a Cluster of Workstations," *Proceedings of the First Symposium on High-Performance Distributed Computing*, pp. 152-160, Syracuse, NY, Sept., 1992.

[15] R. Chin and S. Chanson, "Distributed Object-Based Programming Systems," *ACM Computing Surveys*, pp. 91-127, vol. 23, no. 1, March., 1991.

[16] R. F. Freund and D. S. Cornwell, "Superconcurrency: A form of distributed heterogeneous supercomputing," Supercomputing Review, Vol. 3, Oct. 1990, pp. 47-50.

[17] P. B. Gibbond, "A Stub Generator for Multi-Language RPC in Heterogeneous Environments," *IEEE Trans. Software. Eng. SE*, vol. 13, no. 1, pp. 77-87, January, 1987.

[18] A. S. Grimshaw, D. Shiflet, and A. Nguyen-Tuong, "Campus-Wide Computing: Early Results Using Legion at the University of Virginia," *submitted to Supercomputing '94*.

[19] A. S. Grimshaw, W. A. Wulf, J. C. French, A. C. Weaver, and P. F. Reynolds Jr., "Legion: The Next Logical Step Toward a Nationwide Virtual Computer," University of Virginia, Computer Science TR 94-21, 1994.

[20] A. S. Grimshaw, J. B. Weissman, and W. T. Strayer, "Portable Run-Time Support for Dynamic Object-Oriented Parallel Processing," submitted to *ACM Transactions on Computer Systems*, July, 1993.

[21] A. S. Grimshaw, E. A. West, and W.R. Pearson, "No Pain and Gain! - Experiences with Mentat on Biological Application," *Concurrency: Practice & Experience*, pp. 309-328, Vol. 5, issue 4, June, 1993.

[22] A.S. Grimshaw, J.B.Weissman, E.A. West, and E. Loyot, "Meta Systems: An Approach Combining Parallel Processing And Heterogeneous Distributed Computing Systems," *to appear Journal of Parallel and Distributed Computing.*

[23] A. S. Grimshaw, W. T. Strayer, and P. Narayan, "Dynamic Object-Oriented Parallel Processing," *IEEE Parallel & Distributed Technology: Systems & Applications*, pp. 33-47, May, 1993.

[24] A. S. Grimshaw, "Easy to Use Object-Oriented Parallel Programming with Mentat," *IEEE Computer*, pp.

39-51, May, 1993.

[25] A. S. Grimshaw, and E. Loyot Jr., ELFS: Object-Oriented Extensible File Systems, University of Virginia, Computer Science TR 91-14, 1991.

[26] A. Hac, "Load Balancing in Distributed Systems: A Summary", *Performance Evaluation Review*, ACM, vol. 16, pp. 17-25, February 1989.

[27] P.J. Hatcher, "A Production-Quality C* Compiler for Hypercube Multicomputers," *Proceedings of the Third ACM SIGPLAN Symposium on Principles & Practice of Parallel Programming*, Williamsburg, VA, April 21-24, 1991.

[28] M. Jones, R. F. Rashid, and M. R. Thompson, "An Interface Specification Language for Distributed Processing." *Proceedings of the 12th ACM Symposium on Principles of Programming Languag*es, pp. 225-235, 1985.

[29] J.A. Kaplan and M.L. Nelson, "A Comparison of Queueing, Cluster, and Distributed Computing Systems," NASA Technical Memorandum 109025, NASA LaRC, October, 1993.

[30] Ashfaq Khokhar, et. al., "Heterogeneous Supercomputing: Problems and Issues," *Proceedings of WHP 92 Workshop on Heterogeneous Processing*, IEEE Press, pp. 3-12, Beverly Hills, CA, March, 1992.

[31] J. K. Lee and D. Gannon, "Object Oriented Parallel Programming Experiments and Results," *Proceedings of Supercomputing '91*, pp. 273-282, Albuquerque, NM, 1991.

[32] E. Levy, and A. Silberschatz, "Distributed File Systems: Concepts and Examples," ACM Computing Surveys, vol. 22, No. 4, pp. 321-374, December, 1090.

[33] D. B. Loveman, "High Performance Fortran, "*IEEE Parallel & Distributed Technology: Systems & Applications*, vol. 1, no. 1, pp. 25-42, February, 1993.

[34] F. Manola, S. Heiler, D. Georgakopoulos, M. Hornick, and M. Brodie, "Distributed Object Management," *International Journal of Intelligent and Cooperative Information Systems*, vol. 1, no. 1, June 1992.

[35] J.H. Morris, et al., 'Andrew: A distributed personal computing environment', *Communications of the ACM*, vol. 29, no. 3, March 1986.

[36] R. Mirchandaney, D. Towsley, and J. Stankovic, 'Adaptive Load Sharing in Heterogeneous Distributed Systems,' *Journal of Parallel and Distributed Computing*, Academic Press, no. 9, pp. 331-346, 1990.

[37] N. Nedeljkovic, and M.J. Quinn, "Data-Parallel Programming on a Network of Heterogeneous Workstations," *Proceedings of the First Symposium on High-Performance Distributed Computing*, pp. 28-36, Syracuse, NY, Sept., 1992.

[38] J.R. Nicol, C.T. Wilkes, and F.A. Manola, "Object-Orientation in Heterogeneous Distributed Systems", *IEEE Computer*, vol.26, no. 6., pp. 57-67, June, 1993.

[39] D. Notkin, N., et al.,"Heterogeneous Computing Environments: Report on the ACM SIGOPS Workshop on Accommodating Heterogeneity," *Communications of the ACM*, vol. 30, no. 2, pp. 132-140, February, 1987.

[40] D. Notkin, et al., "Interconnecting Heterogeneous Computer Systems," *Communications of the ACM*, vol. 31, no. 3, pp. 258-273, March, 1988.

[41] S. K. Smith, et al., "Experimental Systems Project at MCC," MCC Technical Report Number: ACA-ESP-089-89, March 2, 1989.

[42] Sun Microsystems. *External Data Representation Reference Manual.* Sun Microsystems, Jan. 1985.

[43] V.S. Sunderam, "PVM: A framework for parallel distributed computing," *Concurrency: Practice and Experience*, vol. 2(4), pp. 315-339, December, 1990.

[44] B. Walker, et al., "The LOCUS Distributed Operating System," *Proceedings of the 9th ACM Symposium on Operating Systems Principles* (Bretton Woods, N. H., Oct.) ACM, New York, 1983.

[45] Min-You Wu, and G.C. Fox, "A Test Suite Approach for Fortran90D Compilers on MIMD Distributed Memory Parallel Computers," *Proceedings of the First Symposium on High-Performance Distributed Computing*, pp. 393-400, Syracuse, NY, Sept., 1992.