2000-04-14

Management System for Heterogeneous Networks Final Report Volume I: Project Summary and Paper

Irvine, Cynthia

Management System for Heterogeneous Networks

Final Report

Volume I: Project Summary and Papers

Naval Postgraduate School Technical Report NPS-CS-00-06

Cynthia E. Irvine
Naval Postgraduate School

H. J. Siegel
Purdue University

Viktor Prasanna
University of Southern California

Debra Hensgen
Formerly with
Naval Postgraduate School

Tim Levin
Anteon Corporation

April 14, 2000

This page is intentionally blank

1 Introduction

The Management System for Heterogeneous Networks (MSHN) project is part of the DARPA/ITO Quorum program. Quorum's goal is to develop technologies to allow mission-critical defense applications to achieve survivable, predictable, and controllable quality of service on a globally managed pool of distributed resources.

The goal of the MSHN Project is to explore the application of adaptive and heuristic matching and scheduling techniques, and modern distributed security methods, to a distributed heterogeneous resource management system (RMS) which allows system resources to be accessed by both MSHN-controlled and external applications. To validate our research and engineering assumptions, a prototype version of MSHN has been developed and demonstrated.

A complete description of the MSHN technical program is found in the research papers which constitute Appendix A. The remainder of this document provides both a high-level overview of the MSHN technical program and a reference guide to the research papers.

The MSHN Project began in 1997, under the direction of Dr. Debra Hensgen. In the fall of 1999, Dr. Cynthia Irvine took on oversight for MSHN. The primary project contract concluded on March 31, 2000. These are the MSHN investigators:

- **Principal Investigators**
  - Dr. Cynthia Irvine, Naval Postgraduate School
  - Richard Freund, Noemix, Inc.
- **Investigators**
  - Dr. Viktor Prasanna, University of Southern California
  - Dr. H.J. Siegel, Purdue University
- **Past Principal Investigators**
  - Dr. Debra Hensgen, formerly with Naval Postgraduate School
  - Dr. Taylor Kidd, formerly with Naval Postgraduate School

2 Architecture

The MSHN design embodies a peer-to-peer architecture [23] composed of the following components:

- Client Library (wrapping each application under MSHN's control)
- Scheduling Advisor (hierarchically replicated)
- Resource Requirements Database (hierarchically replicated)
- Resource Status Server (hierarchically replicated)
- MSHN Daemon (one for each computing resource)
- Application Emulator (at least one for each computing resource)

These components can execute on the same physical machine or can be distributed to reside on separate, heterogeneous machines. The Common Object Request Broker Architecture (CORBA) provides communication between components. Communication security between the MSHN components is provided by the MSHN Security Architecture [57] [56] [22].

The MSHN architecture supports the simultaneous execution of many different client applications, supporting both new and previously encountered applications. MSHN does not assume complete control of its managed resources; rather it allows both MSHN and non-MSHN (viz, non-wrapped) applications to access system resources. Because resources are continuously monitored, external and legacy applications that are not wrapped by the Client Library are accounted for indirectly by their interaction with the system resources.

2.1 Other Architectures

For a comparison with other resource management and heterogeneous computing architectures, see [42]. This work provides background for various aspects of our MSHN work by summarizing relevant papers from a variety of research projects. We include (1) a broad overview of heterogeneous computing (HC); (2) several case studies that give more specific details of applications executing on HC systems; (3) a sampling of current HC tools and environments; (4) methods of classifying HC systems; and (5) techniques for benchmarking machines, techniques for profiling tasks, and schemes that use the information regarding machines and tasks to derive a mapping of the tasks onto the machines.

2.2 Basic System Functions and Attributes

When viewed as a black box, the MSHN system interacts with two actors: applications and resources. MSHN's primary job is assigning and reassigning resources to applications. Included in that functionality is the discovery of resources and the monitoring of both availability of those resources and requirements of the applications that make use of those resources.

2.2.1 Scheduling Advisor Functions

The primary responsibility of the Scheduling Advisor is to determine the best assignment of resources to a set of tasks based on the optimization of a global metric. The Scheduling Advisor depends on the Resource Requirements Database and the Resource Status Server in order to identify an operating point that optimizes the global metric. It responds to scheduling and resource assignment requests from the Client Library. When appropriate, the Scheduling Advisor requests application adaptations via the Client Library. The Scheduling Advisor is also responsible for establishing thresholds to trigger callbacks to the Resource Status Server and Resource Requirements Database (see details below).

2.2.2 Client Library Functions

The Client Library is intended to be linked with both adaptive and non-adaptive applications. It provides the application with a transparent interface to all of the other MSHN components. The Client Library intercepts system calls to collect resource usage and status information, which it forwards to the Resource Requirements Database and the Resource Status Server. The Client Library also intercepts calls that initiate new processes (such as exec()) and consults the Scheduling Advisor for the best place to start that process. It requests execution of applications based on advice from the Scheduling Advisor. Similarly, when notified by the Scheduling Advisor via callbacks, the Client Library can trigger changes to adaptive applications, including the Application Emulator.

2.2.3 Resource Status Server Functions

The role of the Resource Status Server is to maintain a repository of the three types of information about the resources available to MSHN: relatively static (long-term), moderately dynamic (medium-term), and highly dynamic (short-term) information. The Resource Status Server is updated with current data via the Client Library. The Resource Status Server responds to Scheduling Advisor requests with estimates of currently available resources. The Scheduling Advisor sets up callbacks with the Resource Status Server based on resource availability thresholds and Client Library update frequency requirements.

2.2.4 Resource Requirements Database Functions

The Resource Requirements Database is intended to be a repository of information pertaining to the resource usage of applications. The Resource Requirements Database provides this information to the Scheduling Advisor, and it is updated by the Client Library. Callbacks to the Scheduling Advisor are based on either the occurrence of a threshold violation or update frequency requirements.

2.2.5 Daemon (D) Functions

The MSHN Daemon runs on all compute resources available for use by the Scheduling Advisor. It's sole purpose is to start applications as requested by the Client Library.

2.2.6 Application Emulator

The Application Emulator serves two purposes. The first is to simulate applications (that statistically have the same resource usage footprint as the real applications) without the overhead and uncertainty of actually installing, maintaining and running that particular application. The second purpose is to be a resource availability monitor in the absence of any other MSHN-wrapped applications. The daemon starts one instance of the Application Emulator by default at startup. For the purposes of the MSHN demonstration, the Application Emulator functions are performed by a version of the MSHN Daemon.

3 Mapping Algorithms

The mapping (matching and scheduling) research we have conducted was in support of the MSHN Scheduling Advisor. The Scheduling Advisor will include a "toolbox" of mapping techniques from which it can select the most appropriate to use for any given heterogeneous computing and application environment.

3.1 Unified Mapping Framework

We have developed a unified mapping framework for heterogeneous computing systems [3]. Our framework considers multiple types of resources such as compute resources, network resources, I/O devices, and data repositories, such that mapping decisions are based on all the resource requirements. Using our framework, we formulated and studied two novel mapping problems:

- Mapping with advance reservation and data replication
- Mapping with resource co-allocation requirements

In the first problem, we considered the emerging concept of advance reservations where system resources can be reserved in advance for specific time intervals. We assumed that applications with various resource requirements are submitted from participant sites. Each application was assumed to consist of several tasks and was represented by a directed acyclic graph (DAG). The resource requirements were specified at the task level. A task's input data can be data items from its predecessors and/or data sets from data repositories. Input data sets can be accessed from one or more data repositories. A task is ready for execution if all its predecessors have completed, and it has received all the input data needed for its execution. Sources of input data and the execution times of the tasks on various machines along with their availability were considered simultaneously to minimize the overall completion time.

We have developed several heuristic algorithms to solve the above problem. These results are published in [1]. Although we considered multiple resource requirements, tasks were not required to access different types of resources simultaneously.

In the second problem, we considered mapping a set of applications in a heterogeneous computing (HC) system where application tasks require concurrent access to multiple resources of different types. In general, this problem is the resource co-allocation problem. The co-allocation problem can be defined as the problem of simultaneously allocating multiple resources of different types to applications in order to meet specific performance requirements.

We have developed a general framework for mapping with resource co-allocation in HC systems. The framework defined the system and application models and formulated the co-allocation problem. Two graphs were used to represent applications: a directed acyclic graph and a "compatibility graph." The DAG representation is given initially and it stays unchanged throughout the mapping process while the compatibility graph is updated during the mapping process. In classical mapping problems, only DAGs are used to represent the precedence constraints among tasks. In our framework, the co-allocation requirements add another type of constraint among the tasks: resource sharing constraint which is captured by the compatibility graph. Tasks that share one or more resources cannot be executed concurrently due to resource sharing constraints even if they have no precedence constraints among them. Known mapping algorithms for the classical DAG scheduling problem cannot be directly used for the above problem since they consider the precedence constraints only. We have developed heuristic algorithms that can be used with different allocation techniques to efficiently solve the co-allocation problem defined by our framework.

In our approach, multiple DAGs of different applications are combined into a single DAG. All tasks that have satisfied the precedence constraints are ready for allocation provided they have no resource sharing constraints. Using the compatibility graph, we select tasks that can be executed concurrently. This is achieved by finding maximal independent sets in the compatibility graph. These results appear in [2].

3.2 Mapping Heuristics

We have studied heuristics for mapping (viz, scheduling and matching) communicating subtasks to machines in a variety of situations. A genetic algorithm method for static (off-line) mapping of communicating subtasks of a task in a heterogeneous computing (HC) environment is presented in [55]. A way to select from precomputed static mappings, using on-line real-time feedback for automatic target recognition problems is given in [35]. In [41] [38], we describe a hybrid remapper that improves a statically obtained initial mapping by using on-line feedback of run-time execution times of communicating subtasks and machine ready times. A theoretical stochastic model for the mapping of communicating subtasks of a task is presented in [52]. This model is used to show the worth of a greedy approach for mapping heuristics.

We have also considered the case where the tasks to be mapped to machines are independent. Eleven different static mapping heuristics are compared in [15] [14] under several different situations that could occur in a heterogeneous computing environment. This study provides a single basis for comparison and insights into circumstances where one technique will out-perform another. While [15] compares static mapping heuristics, eight dynamic on-line heuristics for mapping a class of independent tasks are compared in [4] [40] [39]. In contrast to static, off-line mappers, which assume a knowledge of what tasks are to be planned for execution during the next day (or other time interval), dynamic, on-line mappers handle tasks as they arrive (without such prior knowledge). Three of the dynamic heuristics compared have been proposed as part of this research. The comparisons show that the selection of a dynamic mapping heuristic in a particular HC environment depends on the arrival rate of tasks and the optimization requirements.

A taxonomy for classifying different matching and scheduling methodologies is given in [16]. This taxonomy may be used to help classify and distinguish the different algorithms available with the MSHN Scheduling Advisor. A framework for simulating different HC environments to allow testing of relative performance of different mapping heuristics under different circumstances is presented in [5]. The paper characterizes an HC environment by using the expected execution times of the tasks that arrive in the system and maps them onto the different machines present in the system.

We contributed a chapter [53] to the upcoming book entitled *Solutions to Parallel and Distributed Computing Problems: Lessons from Biological Sciences*. This chapter summarizes our research that utilized genetic algorithms, including (1) the static use of a genetic algorithm for mapping communicating subtasks, (2) the use of a genetic algorithm to find “off-line mappings to use on-line” in certain environments, and (3) the comparison of eleven different static mapping heuristics (one of which was a genetic algorithm).

We present a summary of our genetic algorithm research for static mappings, our “on-line use of off-line mappings” for the dynamic use of precomputed mappings in certain environments, and the initial stages of our dynamic remapping study in an invited paper [51]. A summary of our genetic algorithm research for static mappings and our dynamic remapping studies is given in the invited keynote paper [43]. An invited journal paper [50] gives a review of some of our earlier mixed-machine HC research. This includes (1) characterization of techniques for mapping tasks on HC systems, (2) the MSHN architecture, and (3) comparisons of various static and dynamic mapping heuristics.

3.3 Performance of Mapping Algorithms

In [6] we studied the performance of four mapping algorithms. The four algorithms include two naive ones: Opportunistic Load Balancing (OLB), and Limited Best Assignment (LBA), and two intelligent greedy algorithms. All of these algorithms, except OLB, use expected run-times to assign jobs to machines. As expected run-times are rarely deterministic in modern networked and server based systems, we first use experimentation and an algorithmic approach [19] to determine some plausible run-time distributions. Using these distributions, we next execute simulations to determine how the mapping algorithms perform. Performance comparisons show that the greedy algorithms produce schedules that, when executed, perform better than naive algorithms, even though the exact run-times are not available to the schedulers. We conclude that the use of intelligent mapping algorithms is beneficial, even when the expected time for completion of a job is not deterministic.

We also performed event simulation experiments to investigate the cost tradeoffs of scheduling jobs in "groups" versus scheduling each job as it arrives [17]. Our results show that if the utilization factor for the system is near 1.0 (viz, when the mean arrival rate is comparable to the total mean service rate of the processors), job grouping is more efficient than per-job scheduling.

4 Resource Modeling and Monitoring

The heart of the Scheduling Advisor component of MSHN is a "model" of the network resources and tasks for which it is responsible. MSHN uses this model to make mapping decisions. The model's data is maintained in the Resource Requirements Database and the Resource Status Server. The effectiveness of MSHN's resource management services depends on how well it can model and monitor its resources and tasks. This section introduces several MSHN Project papers regarding our research into effective techniques for resource modeling and monitoring.

In [34] we determine, through simulation, that providing a more accurate estimate of the network load could permit users of adaptive applications to obtain better performance. We studied the accuracy with which resource loading information, particularly network loading information, must be known in order for applications to successfully, and with agility, adapt [33]. We determine that under many normal conditions, fairly inaccurate estimates of currently available bandwidth suffice. However, when the system is heavily loaded, some strategies can perform much better with very accurate load estimates. The accuracy with which the available bandwidth must be known varies not only with inter-arrival rate, but also with the adaptation strategy used and the percentage of adaptive applications in the system.

In [48] we describe the design, implementation, and results of the first MSHN Client Library prototype. This research develops the mechanism and policy for the Client Library's resource monitoring role and carefully documents how applications can be easily linked with the Client Library. Additionally, we describe a policy for passively gathering network performance characteristics, i.e., latency and throughput, to minimize overhead added to the run-time of test programs.

In [47] we focus on the problem of monitoring the end-to-end performance of adaptive MSHN applications. Based upon a survey of available monitoring tools and analytical experiments, we conclude that the optimal monitoring mechanism: (1) should be passive; (2) should not require domain-specific knowledge of an application; (3) should minimize sources of error; and (4) should have few limitations. No single tool or application component surveyed has all of these characteristics. We describe a new tool whose mechanisms have all of the desired characteristics, and how we implemented it, in detail.

System models that are too detailed incur unnecessary overhead when values corresponding to the detail are being obtained; they are subject to higher variances; and the benefit of computing schedules using them may be outweighed by the time required to compute those schedules. In [18] we propose a model that balances the level of detail, and therefore the quality of their predictions of resource usage, against the cost of computing schedules. To assess the quality of the proposed model, an Application Emulator was designed, built, and used. The results from running the Application Emulator demonstrated that the proposed model is able to predict the relative resource usage of an asynchronous application that has substantially more computation requirements than communication requirements. We refined this model in [49] to correctly estimate the relative execution times of certain communication-intensive, and compute-intensive, asynchronous applications.

As part of our communication scheduling framework we developed an analytical communication model to compute the time for node-to-node communication events [8]. The model represents the network performance between processor pairs using two parameters: start-up cost and data transmission rate. The analytical communication model is represented in a timing diagram, which is input to the scheduling algorithm.

We investigated the capabilities of currently available communication resource status monitoring tools for the purpose of identifying those tools that, with low overhead, can provide accurate, end-to-end communication status information in a Windows NT environment [31]. The techniques used by the various tools are described and the methods for determining the accuracy of these tools are specified.

In [45] we investigated methods of transparently intercept operating system calls made by a robust C4I modeling application, the Extended Air Defense Simulation (EADSIM), to measure the resources required by that application. MSHN utilizes this type of information to determine which version of an application to execute while meeting operational deadlines. We provide the first such data gathered on a complex, contemporary, C4I/air defense model currently in use throughout the DoD, and provide conclusions regarding the trade-offs of computing resources and confidence in simulation outcomes.

5 Distributed Communications

The heterogeneous computing nodes in a metacomputing system are interconnected by several types of networks such as Ethernet, ATM, and FDDI, among others. Many of the metacomputing applications involve frequent and large volumes of data transfer among the nodes. The overall application performance therefore depends largely on the system's communication performance. Network heterogeneity and dynamic run-time variations in network performance present significant challenges for efficient communication.

In the context of such a heterogeneous system, our research addressed the problem of efficient collective communication wherein a group of nodes communicate among one another. We introduced a uniform framework [7] for developing communication schedules for these collective communication patterns. Our framework consisted of analytical models of the heterogeneous network, abstract representations of the communication pattern, and scheduling algorithms. Schedules were adapted at run-time, based on network performance information obtained from a directory service. Our analytical models represented the communication performance between a pair of nodes as the sum of latency and bandwidth components. These components varied from one pair of nodes to another.

Based on this framework we have derived efficient communication schedules for total-exchange [8] [10] cyclic redistribution [9] [11], broadcast, and multicast [12] [13]. Our scheduling algorithms incorporated techniques from bi-partite graph matching, spanning tree algorithms, and shop scheduling theory. For the total-exchange problem, the open shop algorithm developed schedules which had a bounded completion time of at most twice the optimal. For this problem, our simulation results showed performance improvements of up to a factor of 5 over previous approaches. For the cyclic redistribution problem, we have implemented the open shop algorithm on a Cray T3E. Our results showed consistent performance improvements of up to 60 percent, compared with a baseline algorithm. Our scheduling techniques for the broadcast and multicast problems were based on spanning tree algorithms. Performance improvements of over a factor of 10 were achieved.

6 Performance Metrics

In a distributed heterogeneous computing environment, users' tasks are allocated resources to simultaneously satisfy, to varying degrees, the tasks' different, and possibly conflicting, quality of service (QoS) requirements. When the total demand placed on system resources by the tasks, for a given interval of time, exceeds the resources available, some tasks will receive degraded service or no service at all. One part of a measure to quantify the success of a resource management system (RMS) in such a distributed environment is the collective value of the tasks completed during an interval of time, as perceived by the user, application, or policy maker. The Flexible Integrated System Capability (FISC) ratio introduced in [32] is a measure for quantifying this collective value. The FISC ratio is a multi-dimensional measure, and may include priorities, versions of a task or data, deadlines, situational mode, security, application- and domain-specific QoS, and dependencies. In addition to being used for evaluating and comparing RMSs, the FISC ratio can be incorporated as part of the objective function in a system's scheduling heuristics.

7 Security

The MSHN security architecture [57] [56] [22] is based upon separation of services into four distinct partially ordered privilege domains, and provides security support for authentication, communications security, access control and accountability. It is designed to take advantage of operating system support for domains, where available, and uses emerging public key technology as an nearterm (interim) solution.

A method for articulating network security functional requirements, and for measuring their fulfillment, is presented in [27] [37]. Using this method, security in a quality of service framework (QoSS) is discussed in terms of variant security mechanisms and dynamic security policies. It is also shown how QoSS can be represented in a network scheduler benefit function. Fundamental QoSS concepts are discussed in [28].

In [29] we present an analysis of the layered and variable security services and requirements presented to a resource management system. We provide a network system model for analyzing how user and application choices and limits can affect the overall security provided by the RMS. We also present a method for fairly measuring the effectiveness of an RMS in performing security allocation and assignments with respect to security choices made by metacomputer users and applications

To knowledgeably assign computing and network resources to tasks, the resource management system (RMS) needs to know the resource-utilization costs associated with various network security services which it may assign to tasks. In [25] [26] we define a preliminary security service taxonomy defining the range of security services an RMS may need to manage; utilizing this taxonomy, we then provide a framework for defining the costs associated with network security services.

In [24] we address the problem of how users and administrators can understand and easily interact with a wide range of security services and mechanisms. We provide method for translation of a simplified user abstraction of security to detailed underlying mechanisms, such that users can be presented with a coherent user-level view of available security options.

We describe an approach for representing the level of resources consumed by jobs under the control of a resource management system [36], and it is shown how this measurement of resource usage can be combined with a notion of user preferences to reflect a restrictive resource-usage policy for network management.

8 Demonstration/Implementation

The MSHN Prototype consists of several inter-communicating components [23], the functions of which are described in the various MSHN documents. Development of the MSHN Scheduling Advisor component occurred at the Noemix site, and the other components were developed at NPS. Component integration was supported by both Noemix and NPS. In the Fall of 1999 all of the MSHN Prototype development and integration was transferred to the Noemix site. The following MSHN papers and theses describe various implementation issues regarding the prototype demonstration: use of CORBA, [20] [46] [21] real time support [44] system specification using UML [30] and Java threads [54].

9 Future Directions

9.1 Mapping Algorithms

Mapping research in progress builds on our past studies and results. We are developing techniques for mapping tasks to machines in heterogeneous environments where tasks have priorities, multiple versions, and deadlines. We are using a subset of FISC as the performance measure. We are designing two static mappers, selecting two that performed very well in our previous studies. After this is completed, possible future work would involve developing dynamic (on-line) mappers for such tasks and performance measure. We could also extend the performance measure to include security and application QoS attributes.

9.2 Security

The security architecture of the MSHN project may be applicable to other RMS architectures and to selected DoD applications. Additional work will be needed to understand how commercially available security architectures can be generalized for RMS support. The extension and development of the notion of Quality of Security Service is another area for further research. Theoretical work is needed to understand how QoSS can embrace survivability notions. The QoSS development of the MSHN project needs to be refined and applied to a variety of scheduling frameworks.

10 Acknowledgements

The participants in the MSHN effort are grateful for the support and guidance we have received from Dr. Gary Koob and his staff for the DARPA/ITO Quorum program. We wish to thank the many students, staff, and investigators at our respective institutions and the many other Quorum investigators who have provided us with insights and encouragement.

References

[1] Ammar Alhusaini, Viktor Prasanna, and C. S. Raghavendra. A Unified Resource Scheduling Framework for Heterogeneous Computing Environments. In *Proceedings of the Eighth Heterogeneous Computing Workshop (HCW '99)*, pages 156–165, San Juan, PR, April 1999.

[2] Ammar Alhusaini, Viktor Prasanna, and C. S. Raghavendra. A Framework For Mapping With Resource Co-Allocation In Heterogeneous Computing Systems. In *Proceedings of the Ninth Heterogeneous Computing Workshop (HCW 2000)*, Cancun, Mexico, To appear May, 2000.

[3] Ammar H. Alhusaini. *A Unified Mapping Framework For Heterogeneous Computing Systems*. PhD thesis, University of Southern California, Los Angeles, CA, In Progress 2000.

[4] Shoukat Ali. A comparative study of dynamic mapping heuristics for a class of independent tasks onto heterogeneous computing systems. M. S. thesis, Purdue University, West Lafayette, IN, August 1999.

[5] Shoukat Ali, Howard Jay Siegel, Muthucumaru Maheswaran, Debra Hensgen, and Sahra Ali. Task Execution Time Modeling for Heterogeneous Computing Systems. In *Proceedings of the Ninth Heterogeneous Computing Workshop (HCW 2000)*, Cancun, Mexico, To appear May, 2000.

[6] Robert Armstrong, Debra Hensgen, and Taylor Kidd. The Relative Performance of Various Mapping Algorithms is Independent of Sizable Variances in Run-Time Predictions. In *Proceedings of the Seventh Heterogeneous Computing Workshop (HCW '98)*, pages 79–87, Orlando, FL, March 1998.

[7] Prashanth B. Bhat. *Communication Scheduling Techniques For Distributed Heterogeneous Systems*. Ph. D. thesis, University Of Southern California, Los Angeles, CA, August 1999.

[8] Prashanth B. Bhat, Viktor K. Prasanna, and C. S. Raghavendra. Adaptive Communication Algorithms for Distributed Heterogeneous Systems. In *Proceedings of the 7th International Symposium On High Performance Distributed Computing (HPDC-7 '98)*, pages 310–321, Chicago, ILL, July 1998.

[9] Prashanth B. Bhat, Viktor K. Prasanna, and C. S. Raghavendra. Block-Cyclic Redistribution over Heterogeneous Networks. In *Proceedings of the 11th International Conference On Parallel And Distributed Computing Systems (PDCS '98)*, pages 242–249, Chicago, ILL, September 1998.

[10] Prashanth B. Bhat, Viktor K. Prasanna, and C. S. Raghavendra. Adaptive Communication Algorithms for Distributed Heterogeneous Systems. *Journal of Parallel and Distributed Computing*, 59(2):252–279, November 1999.

[11] Prashanth B. Bhat, Viktor K. Prasanna, and C. S. Raghavendra. Block-Cyclic Redistribution over Heterogeneous Networks. *Accepted for publication in Cluster Computing: The Journal of Networks, Software Tools and Applications*, To appear 2000.

[12] Prashanth B. Bhat, C. S. Raghavendra, and Viktor K. Prasanna. Efficient Collective Communication in Distributed Heterogeneous Systems. In *Proceedings of the 19th International Conference On Distributed Computing Systems*, pages 15–24, Austin, TX, May 1999.

[13] Prashanth B. Bhat, C. S. Raghavendra, and Viktor K. Prasanna. Efficient Collective Communication in Distributed Heterogeneous Systems. *Submitted to the Journal of Parallel and Distributed Computing*, To appear 2000.

[14] Tracy D. Braun, Howard Jay Siegel, Noah Beck, Ladidlau L. Boloni, Muthucumaru Maheswaran, Albert I. Reuther, James P. Robertson, Mitchell D. Theys, Bin Yao, Debra Hensgen, and Richard F. Freund. A Comparison Study of Eleven Static Heuristics for Mapping a Class of Independent Tasks onto Heterogeneous Distributed Computing Systems. Technical Report TR.ECE 00-4, Purdue University, West Lafayette, IN, March 2000.

[15] Tracy D. Braun, Howard Jay Siegel, Noah Beck, Ladislau L. Boloni, Muthucumaru Maheswaran, Albert I. Reuther, James P. Robertson, Mitchell D Theys, Bin Yao, Debra Hensgen, and Richard F. Freund. A Comparison Study of Static Mapping Heuristics for a Class of Meta-Tasks on Heterogeneous Computing Systems. In *Proceedings of the Eighth Heterogeneous Computing Workshop (HCW '99)*, pages 15–29, San Juan, Puerto Rico, April 1999.

[16] Tracy D. Braun, Howard Jay Siegel, Noah Beck, Ladislau L. Boloni, Muthucumaru Maheswaran, Albert I. Reuther, James P. Robertson, Mitchell D Theys, and Bin Yao. A Taxonomy for Describing Matching and Scheduling Heuristics for Mixed-Machine Heterogeneous Computing Systems. In *Proceedings of the IEEE Workshop on Advances in Parallel and Distributed Systems*, pages 330–335, West Lafayette, IN, October 1998.

[17] James M. Breitinger. Optimal size of job pool for initiating a scheduling event. M. S. thesis, Naval Postgraduate School, Monterey, CA, September 1999.

[18] Paul F. Carff. When is a simple model adequate for use in scheduling in mshn. M. S. thesis, Naval Postgraduate School, Monterey, CA, March 1999.

[19] Thomas S. Cook. Dynamically determining distribution statistics in a distributed environment. M. S. thesis, Naval Postgraduate School, Monterey, CA, December 1999.

[20] Alpay Duman. The use and run-time overhead of corba in mshn project. M. S. thesis, Naval Postgraduate School, Monterey, CA, September 1998.

[21] Alpay Duman, Debra Hensgen, David St. John, and Taylor Kidd. Are CORBA Services Ready to Support Resource Management Middleware for Heterogeneous Computing. In *Proceedings of the Eighth Heterogeneous Computing Workshop(HCW '99)*, pages 83–96, San Juan, Puerto Rico, April 1999.

[22] John English. A management system for heterogeneous networks (mshn) security analysis. M. S. thesis, Naval Postgraduate School, Monterey, CA, September 1997.

[23] Debra A. Hensgen, Taylor Kidd, David St. John, Matthew C. Schnaidt, Howard Jay Siegel, Tracy D. Braun, Muthucumaru Maheswaran, Shoukat Ali, Jong-Kook Kim, Cynthia Irvine, Tim Levin, Richard F. Freund, Matt Kussow, Michael Godfrey, Alpay Duman, Paul Carff, Shirley Kidd, Viktor Prasanna, Prashanth Bhat, and Ammar Alhusaini. An Overview of MSHN: The Management System for Heterogeneous Networks. In *Proceedings of the Eighth Heterogeneous Computing Workshop (HCW '99)*, pages 184–198, San Juan, Puerto Rico, April 1999.

[24] Cynthia Irvine and Tim Levin. A Note on Mapping User-Oriented Security Policies to Complex Mechanisms and Services. Technical Report NPS-CS-99-08, Naval Postgraduate School, Monterey, CA, June 1999.

[25] Cynthia Irvine and Tim Levin. Toward a Taxonomy and Costing Method for Security Services. In *Proceedings of the Computer Security Applications Conference*, pages 183–188, Phoenix, AZ, December 1999.

[26] Cynthia Irvine and Tim Levin. Toward a Taxonomy and Costing Method for Security Services. Technical Report NPS-CS-99-07, Naval Postgraduate School, Monterey, CA, June 1999.

[27] Cynthia Irvine and Tim Levin. An Introduction to Quality of Security Services. Technical Report NPS-CS-00-005, Naval Postgraduate School, Monterey, CA, April 2000.

[28] Cynthia Irvine and Tim Levin. The Effects of Security Choices and Limits in a Metacomputing Environment. Technical Report NPS-CS-00-004, Naval Postgraduate School, Monterey, CA, April 2000.

[29] Cynthia Irvine and Tim Levin. Toward Quality of Service in a Resource Management System Benefit Function. In *Proceedings of the Ninth Heterogeneous Computing Workshop (HCW 2000)*, Cancun, Mexico, To appear May, 2000.

[30] David St. John, Taylor Kidd, Debra Hensgen, Mantak Shing, and Shirley Kidd. Experiences Using Semi-Formal Methods During Development of Distributed, Research-Oriented, System-Level Software. Technical Report NPS-CS-99-04, Naval Postgraduate School, Monterey, CA, May 1999.

[31] Ronald Jacobs Jr. Identifying accurate resource monitoring tools and techniques. M. S. thesis, Naval Postgraduate School, Monterey, CA, September 1999.

[32] Jong-Kook Kim, Debra A. Hensgen, Taylor Kidd, Howard Jay Siegel, David St. John, Cynthia Irvine, Tim Levin, N. Wayne Porter, Viktor K. Prasanna, and Richard F. Freund. A QoS Performance Measure Framework for Distributed Heterogeneous Networks. In *Proceedings of the Eighth Euromicro Workshop on Parallel and Distributed Processing*, pages 18–27, Rhodes, Greece, January 2000.

[33] John Kresho, Debra Hensgen, Taylor Kidd, and Geoffrey Xie. Determining the Accuracy Required in Resource Load Prediction to Successfully Support Application Agility. In *Proceedings of the Second IASTED European Parallel and Distributed Systems Conference (Euro-PDS 98)*, pages 224–254, Vienna, Austria, July 1998.

[34] John P. Kresho. Quality network load information improves performance of adaptive applications. M. S. thesis, Naval Postgraduate School, Monterey, CA, September 1997.

[35] Yu-Kwong Kwok, Anthony A. Maciejewski, Howard Jay Siegel, Arif Ghafoor, and Ishfaq Ahmand. Evaluation of a Semi-Static Approach to Mapping Dynamic Iterative Tasks onto Heterogeneous Computing Systems. In

Proceedings of the International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN '99), pages 204–209, Fremantle, Australia, June 1999.

[36] Tim Levin and Cynthia Irvine. An Approach to Characterizing Resource Usage and User Preferences in Benefit Functions. Technical Report NPS-CS-99-005, Naval Postgraduate School, Monterey, CA, June 1999.

[37] Tim Levin and Cynthia Irvine. Quality of Security Service in a Resource Management System Benefit Function. Technical Report NPS-CS-00-02, Naval Postgraduate School, Monterey, CA, November 1999.

[38] Muthucumaru Maheswaran. *Software Issues On Mapping Applications Onto Heterogeneous Machines And The Performance Of Krylov Algorithms On Parallel Machines*. Ph. D. thesis, Purdue University, West Lafayette, IN, December 1998.

[39] Muthucumaru Maheswaran, Shaoukat Ali, Howard Jay Siegel, Debra Hensgen, and Richard F. Freund. Dynamic Matching and Scheduling of a Class of Independent Tasks onto Heterogeneous Computing Systems. In *Proceedings of the Eighth Heterogeneous Computing Workshop (HCW '99)*, pages 30–44, San Juan, Puerto Rico, April 1999.

[40] Muthucumaru Maheswaran, Shoukat Ali, Howard Jay Siegel, Debra Hensgen, and Richard F. Freund. Dynamic Mapping of a Class of Independent Tasks onto Heterogeneous Computing Systems. *Journal of Parallel and Distributed Computing*, 59(2):107–131, November 1999.

[41] Muthucumaru Maheswaran, Tracy Braun, and Howard Jay Siegel. High Performance Mixed-Machine Heterogeneous Computing. In *Proceedings of the Sixth Euromicro Workshop on Parallel and Distributed Processing*, pages 3–9, Madrid, Spain, January 1998.

[42] Muthucumaru Maheswaran, Tracy D. Braun, and Howard Jay Siegel. *Heterogeneous Distributed Computing, Encyclopedia of Electrical and Electronics Engineering*, volume 8, pages 679–690. John Wiley and Sons, Inc., New York, NY, 1999.

[43] Muthucumaru Maheswaran and Howard Jay Siegel. A Dynamic Matching and Scheduling Algorithm for Heterogeneous Computing Systems. In *Proceedings of the Seventh Heterogeneous Computing Workshop (HCW '98)*, pages 57–69, Orlando, FL, March 1998.

[44] Panagiotis Papadatos. Implementation of real-time mshn using ace and tao. M. S. thesis, Naval Postgraduate School, Monterey, CA, September 1999.

[45] N. Wayne Porter. Resource usage for adaptive c4i models in a heterogeneous computing environment. M. S. thesis, Naval Postgraduate School, Monterey, CA, June 1999.

[46] Matt Schnaidt, Alpay Duman, and Ted Lewis. A Comparison of C++ Sockets and Corba in a Distributed Matrix Multiply Application. Technical Report NPS-CS-99-001, Naval Postgraduate School, Monterey, CA, June 1998.

[47] Matt Schnaidt, Debra Hensgen, John Falby, Taylor Kidd, and David St. John. Passive, Domain-Independent, End-to-End Message Passing Performance Monitoring to Support Adaptive Applications in MSHN. In *Proceedings of the Eighth International Symposium in High Performance Distributed Computing*, pages 3–9, Redondo Beach, CA, August 1999.

[48] Matthew C. L. Schnaidt. Design, implementation, and testing of mshn's application resource monitoring library. M. S. thesis, Naval Postgraduate School, Monterey, CA, December 1998.

[49] Blanca A. Shaeffer. Refining a task-execution time prediction model for use in mshn. M. S. thesis, Naval Postgraduate School, Monterey, CA, March 2000.

[50] Howard Jay Siegel and Shoukat Ali. Techniques for Mapping Tasks to Machines in Heterogeneous Computing Systems. *Journal of Systems Architecture, Special Issue on Heterogeneous Distributed and Parallel Architectures: Hardware, Software and Design Tools*, To appear 2000.

[51] Howard Jay Siegel and Muthucumaru Maheswaran. Mapping Tasks onto Heterogeneous Computing Systems. In *Proceedings of the IX Simposio Brasileiro de Arquitetura de Computadores - Processamento de Alto Desempenho (IX Brazilian Symposium on Computer Architectures - High Performance Computing)*, pages 3–17, Sao Paulo, Brazil, October 1997.

[52] Min Tan and Howard Jay Siegel. A Stochastic Model for Heterogeneous Computing and Its Application in Data Relocation Scheme Development. *IEEE Transactions on Parallel and Distributed Systems*, 9(11):1088–1101, November 1998.

[53] Mitchell D. Theys, Tracy D. Braun, Yu-Kwong Kwok, Howard Jay Siegel, and Anthony A. Maciejewski. *Mapping of Tasks onto Distributed Heterogeneous Computing Systems Using a Genetic Algorithm Approach, in Solutions to Parallel and Distributed Computing Problems: Lessons From Biological Sciences*. John Wiley and Sons, Inc., To appear 2000.

[54] Pedro Tsai. Re-targeting the graze performance debugging tool for java threads and analyzing the re-targeting to automatically parallelized (fortran) code. M. S. thesis, Naval Postgraduate School, Monterey, CA, March 2000.

[55] Lee Wang, Howard Jay Siegel, Vwani P. Roychowdhury, and Anthony A. Maciejewski. Task Matching and Scheduling in Heterogeneous Computing Environments Using a Genetic-Algorithm-Based Approach. *Journal of Parallel and Distributed Computing*, 47(1):8–22, November 1997.

[56] Roger Wright, David J. Shifflett, and Cynthia E. Irvine. Security Architecture for a Virtual Heterogeneous Machine. In *Proceedings of the Fourteenth Computer Security Applications Conference*, pages 167–177, Phoenix, AZ, December 1998.

[57] Roger E. Wright. Integrity architecture and security services demonstration for management system for heterogeneous networks. M. S. thesis, Naval Postgraduate School, Monterey, CA, September 1998.