Dagstuhl Seminar No. 01241

Management of Metacomputers

June 10 – 15, 2001

Organized by

Francine D. Berman, UCSD, La Jolla, USA
Alexander Reinefeld, ZIB and HU, Berlin, Germany
Uwe Schwiegelshohn, Universität Dortmund, Germany

Contents

1 Preface

1.1 Background and Motivation

The success of the Internet along with the worldwide growing number of high-performance computers has led to the concepts of *metacomputing* or, more recently, *computational grids*. In principle, a metacomputer can be considered as an extension of a distributed computer with a variety of geographically dispersed resources, such as supercomputers, storage systems, data sources and special devices. Ideally, such a metacomputer is seen as a single unified resource by the user. It typically consists of various architectures with different application software. Those architectures usually belong to different owners and are accessed from a large number of independent users.

As already mentioned the distributed nature of an ideal metacomputer environment is transparent to the user, that is, he or she only needs to describe the constraints connected with a job while the system selects the most suitable machine for the execution of this job. This selection process may be subject to a large variety of different constraints including access restrictions, user priorities, machine workload, job characteristics and user preferences. In addition, the metacomputer structure may change due to maintenance shut downs or temporary failures of sub-systems. It is the task of the management software to handle those problems, that is, to provide the desired transparent access to the users while at the same time considering any special requests from users and owners. Therefore, the management software is a key component of a metacomputer.

1.2 Contents of the Seminar

The architecture and the methods of such a management software were the focus of this Dagstuhl seminar. The participants explored and analyzed the design, implementation, and deployment of metacomputer management systems. Eighteen talks were given on the various aspects and the state-of-the-art of metacomputer management.

The first day of the seminar (Monday) was devoted to the topic of scheduling. Various projects, concepts, and new approaches have been presented by

Jon Weissman, Dick Epema, Uwe Schwiegelshohn, Larry Carter, and Volker Sander. The day was concluded with a discussion on current aspects and future developments in the field.

The talks of Marian Bubak (given by Roland Wismüller), Barton Miller, and Arnaud Legrand on Tuesday focused on performance issues, program development and security issues in grid environments. In the afternoon, Lennart Johnsson discussed aspects of grid application tools, and thereafter several large scale projects were briefly presented by Volker Sander (Globus), Thilo Kielmann (DAS), Florian Schintke (Datagrid), André Merzky (Cactus), and Steve Chapin (Legion). The pros and cons of these projects were compared and vividly discussed until late in the evening.

On the third day, various topics of grid infrastructure, grid components, and application specific grids were presented by André Merzky, Thilo Kielmann, and Domenico Talia. In the afternoon, participants took a hike through the nearby forests, discussing various research topics in a leisurely surrounding at an excellent weather.

Thursday was filled with talks on grid infrastructure and tools. Steve Chapin, Ramin Yahyapour, Jean-Marc Nicod, Alexander Reinefeld, Roland Wism"uller, Florian Schintke, and Volker Lindenstruth presented their work on the various components used to build metacomputer environments. Again, some of these topics were discussed in more detail until late in the evening.

1.3 Character of the Seminar

Schloss Dagstuhl was a perfect environment to cover a wide range of aspects of metacomputer management. Due to the small group size, there was no pressure to remind speakers to end their talks in time, which made the organization very easy. We ended up with each speaker taking about an hour including questions and answers. In such an open atmosphere, the discussion was often quite vivid and provided useful hints for further research. The seminar benefited by a wide spectrum of participants ranging from young researchers just starting their careers up to full professors who have worked for already a long time in the field.

Arguing from the very positive reactions of the participants, the seminar was perceived as a very fruitful event. Staying one week together at the beautiful place of Dagstuhl has stimulated the scientific and private exchange between the international participants.

Unfortunately, one of the co-organizers, Francine Berman, was unable to attend, because just shortly before the start of the seminar she took over the highly reputed, but also very demanding job as the director of the San Diego Supercomputing Center (SDSC).

2 Final Program

Monday, 11 June 2001

09:00 - 10:00	*Introduction by the Organizers* Uwe Schwiegelshohn, Alexander Reinefeld

Session: Scheduling, Advance Reservation

10:00 - 11:00	Jon B. Weissman, University of Minnesota *Scheduling Computational Grids: A Five-Year Retrospective*
11:00 - 12:00	Dick H.J. Epema, TU Delft - *The Influence of Communication on the Performance of Co-Allocation* - *Flocking as a Paradigm for Discovering and Sharing Resources in Grids*
12:15 - 14:00	Lunch
14:00 - 15:00	Uwe Schwiegelshohn, Universität Dortmund *Evaluation of Scheduling Algorithms for Grid Computing*
15:00 - 16:00	Larry Carter, Univ. of California - San Diego *Bandwidth-centric scheduling of independent tasks on a heterogeneous grid*
16:00 - 16:30	Coffee and cookies
16:30 - 17:45	Volker Sander, KFA Jülich *GARA: Advance Reservation Architecture and API*
18:00	Dinner
19:00 - 21:00	*Discussion on Grid Scheduling and Security* Moderators: Uwe Schwiegelshohn, Alexander Reinefeld

Tuesday, 12 June 2001

Performance Issues, Program Development

09:00 - 10:00	Marian Bubak, AGH & Acad. Comp. Center - Krakow (talk given by Roland Wismüller) *Proposal of the Tool Support for Grid Application Monitoring*
10:00	Coffee
10:15 - 11:15	Barton P. Miller, Univ. Wisconsin - Madison *Playing Inside the Blackbox: Using Dynamic Instrumentation to Create Security Holes*
11:15 - 12:15	Arnaud Legrand, ENS - Lyon *Heterogeneity Considered Harmful to Algorithm Designers*
12:15 - 14:00	Lunch
14:00 - 15:00	Lennart Johnsson *Grid Application Development - The Ultimate Challenge for Code Portability and Adaptivity*
15:00 - 15:30	Coffee and cookies
15:30 - 18:00	*Presentations and Discussion on Large-Scale Grid Testbeds* Moderators: Volker Sander (Globus), Thilo Kielmann (DAS), Florian Schintke (DataGrid), Steve Chapin (Legion), André Merzky (Cactus)
18:00	Dinner

Wednesday, 13 June 2001

Grid Infrastructure

09:00 - 10:00	André Merzky, ZIB Berlin *Thoughts about Data Management in Grid Environments*
10:00 - 10:15	Coffee
10:15 - 11:15	Thilo Kielmann, Vrije Universiteit Amsterdam *Grid-aware Communication Components*
11:15 - 12:15	Domenico Talia, ISI-CNR *Knowledge Discovery and Data Mining in Grid-Based Distributed Environments*
12:15 - 14:00	Lunch
14:00 - 18:00	Hike
18:00	Dinner

Thursday, 14 June 2001

Grid Infrastructure (contd.)

09:00 - 10:00	Steve Chapin, Syracuse University *Internet2/Qbone Bandwidth Broker*
10:00 - 10:15	Coffee
10:15 - 11:15	Ramin Yahyapour, Universität Dortmund *NWIRE Resource-Management-System*
11:15 - 12:15	Jean-Marc Nicod, Besançon *Computational Servers in a Metacomputing Environment*
12:15 - 14:00	Lunch

Metacomputer Components

14:00 - 15:00	Alexander Reinefeld, ZIB and HU Berlin *Integrating Local Resources into the Grid: The Resource Specification Problem*
15:00 - 15:45	Roland Wismüller, TU Munich *Monitoring and Performance Analysis in Distributed Systems*
15:45 - 16:15	Coffee and cookies
16:15 - 18:00	Florian Schintke, ZIB Berlin; Lord Hess, Volker Lindenstruth, KIP Heidelberg *How to Build a Reliable Cluster for the Grid*
18:00	Dinner

3 Abstracts of the Presentations

3.1 Scheduling Computational Grids: A Five-Year Retrospective

Jon B. Weissman
Department of Computer Science University of Minnesota, Twin Cities

This talk reflects our recent experience in scheduling computational Grids over the past five years. We begin by defining the space of schedulers along four dimensions: application model, scheduling type, performance metrics, and Grid assumptions. We then present several scheduling models achieved by making ”vertical slices” across this space, implementations of these models, and performance results. In particular, we will present schedulers for data-parallel applications in both LAN and WAN environments, and a metascheduler that can schedule multi-site meta-applications, both guided by a completion time metric. We conclude with thoughts relating to a new class of schedulers based on coordination of different application schedulers.

3.2 Co-Allocation in Wide-Area Systems

Dick Epema
Delft University of Technology

In order to benefit from computing capacity available in multiple locations, large applications may require co-allocation, that is, the simultaneous assignment of processors under the management of multiple independent resource managers. An important impediment to having parallel applications run in this way may be the relatively slow wide-area connections. We present simulation results for co-allocation in wide-area systems modelled after our Distributed ASCI Supercomputer for varying speed ratios between local and wide-area links, and for different request types made by parallel jobs. These types range from simply stating the total number of processors required, to stating the precise numbers of processors needed in each of the participating sites.

In addition, we present some ideas for extending our previous work on flocking in Condor to resource discovery for co-allocation. In particular, rather than having co-allocation actions resemble database transactions with two-phase commit, which has been advocated elsewhere, we propose an optimistic scheme in which a local resource manager exchanges information on resource availability with other sites and takes co-allocation decisions on its own, and in which applications subsequently co-claim the tentatively assigned resources.

This is joint work with Anca Bucur, also from Delft University.

3.3 Evaluation of Scheduling Algorithms for Grid Computing

Uwe Schwiegelshohn
Computer Engineering Institute, University Dortmund

A necessary condition for computational grids is the willingness of a sufficient number of resource owners to provide their resources for the grid. As most resources are not exclusively used in the grid this will only happen if other (local) tasks of these machines are not significantly affected by the grid. This requires grid scheduling to especially consider policy constraints of the various owners. In the talk we explain a general path to derive first an objective function and then a scheduling algorithm from such a policy. This path is based on the existence of at least one typical user workload and uses multicriteria optimization and algorithmic evaluation with simulation. However, due to additional algorithmic constraints, like, for instance, online requirements, and due to differences between various available workloads the development of a suitable scheduling algorithm is an iterative process. Moreover, it is assumed that the workload will not be affected by the choice of the scheduling algorithm. This assumption may not be true as the user behavior is typically dependent on the circumstances of which the scheduling algorithm is an important part. Therefore, we suggest to include the user behavior into the simulation and evaluation model.

3.4 Bandwidth-centric allocation of independent tasks on heterogeneous platforms

Larry Carter
University of California at San Diego
Joint work with Olivier Beaumont, Arnaud Legrand and Yves Robert (of ENS, Lyon) and Jeanne Ferrante (UCSD)

We consider the problem of allocating a large number of independent, equal-sized tasks to a heterogenerous "grid" computing platform. Such problems arise in collaborative computing efforts like SETI@home. We use a tree to model a grid, where resources can have different speeds of computation and communication, as well as different overlap capabilities. We define a base model, and show how to determine the maximum steady-state throughput of a node in the base model, assuming we already know the throughput of the subtrees rooted at the node's children. Thus, a bottom-up traversal of the tree determines the rate at which tasks can be processed in the full tree. The best allocation is 'bandwidth-centric': if enough bandwidth is available, then all nodes are kept busy; if bandwidth is limited, then tasks should be allocated only to the children which have sufficiently small communication times, regardless of their computation power.

We then show how nodes with other capabilities - ones that allow more or less overlapping of computation and communication than the base model - can be transformed to equivalent nodes in the base model. We also show how to handle a more general communication model.

Finally, we present simulation results of several demand-driven task allocation policies that show that our bandwidth-centric method obtains better results than allocating tasks to all processors on a first-come, first serve basis.

3.5 GARA: Advance Reservation Architecture and API

Volker Sander
Zentralinstitut für Angewandte Mathematik, Forschungszentrum Jülich GmbH

High-end networked applications such as distance visualization, distributed data analysis, and advanced collaborative environments have demanding quality of service (QoS) requirements. Particular challenges include concurrent flows with different QoS specifications, high bandwidth flows, application-level monitoring and control, and end-to-end QoS across networks and other devices. The presentation describes a QoS architecture and implementation that together help to address these challenges. The General-purpose Architecture for Reservation and Allocation (GARA) supports flow-specific QoS specification, immediate and advance reservation, and online monitoring and control of both individual resources and heterogeneous resource ensembles. Mechanisms provided by the Globus toolkit are used to address resource discovery and security issues when resources span multiple administrative domains. The prototype implementation builds on differentiated service mechanisms to enable the coordinated management of two distinct flow types—foreground media flows and background bulk transfers—as well as the co-reservation of networks, CPUs, and storage systems. To demonstrate GARA's ability to deliver advance reservations for multiple Grid resources, several experimental results obtained on a differentiated services testbed will be presented.

3.6 Proposal of the Tool Support for Grid Application Monitoring

Marian Bubak
Institute of Computer Science AGH, Krakow, Poland Academic Computer Centre – CYFRONET AGH, Krakow, Poland

This talk presents a concept of a tool environment for application monitoring on the Grid. Based on the different layers of the grid architecture, the data to be monitored on each layer is identified. An architecture for a monitoring system is proposed and a communication protocol between the different parts of the monitoring environment is discussed. We also focus on performance issues concerning the gathering of monitoring data and the scalability of the monitoring system.

3.7 Playing Inside the Blackbox: Using Dynamic Instrumentation to Create Security Holes

Barton P. Miller
Computer Sciences Department, University of Wisconsin

Programs in execution have long been considered to be immutable objects. Object code and libraries are emitted by the compiler, linked and then executed; any changes to the program require revisiting the compile or link steps. In contrast, we consider a running program to be an object that can be examined, instrumented, and re-arranged on the fly. The DynInst API provides a portable library for tool builders to construct tools that operate on a running program. Where previous tools might have required a special compiler, linker, or run-time library, tools based on DynInst can operate directly on unmodified binary programs during execution. In this talk, I will discuss how this technology can be used to subvert system security. The discussions will be based on two example cases: bypassing access to a license server (in Framemaker) and exposing vulnerabilities in a distributed scheduling system (in Condor).

For the license-server study, we constructed a collection of Dyninst-based tools that allowed us to understand the control flow within the application (Framemaker) program. As a result, we were able to detect and remove Framemaker's contact with the license server. In addition, there are frequent checks within Framemaker to see if it has cached a valid license credential. Using our Dyninst-based tools, we were able to locate and neutralize this check.

For the Condor study, we created "lurker" processes that can be left latent on a host in the Condor pool. These lurker processes lie in wait for subsequent Condor jobs to arrive on the infected host. The lurker will then use Dyninst to attach to the newly-arrived victim job and take control. Once in control, the lurker can cause the victim job to make requests back to its home host, causing it execute almost any system call it would like.

For each of these cases, we provide suggestions as to how to make them less vulnerable to attack.

3.8 Heterogeneity Considered Harmful to Algorithm Designers

Arnaud Legrand
Ecole Normale Supérieure de Lyon, Laboratoire LIP,
Projet CNRS-INRIA ReMaP

We present some algorithmic issues on heterogeneous platforms and more particularly load balancing problems raised by the implementation of dense linear algebra kernels on heterogeneous platforms. We show that on heterogeneous platforms, data dependencies, communication costs and control overhead severely impact classical dynamic schemes. Static strategies suppress (or at least minimize) data redistributions and memory management overhead while preserving parallelism but they cannot be used when some resources are shared. Anyway, on heterogeneous dedicated platforms, data distribution must obey a much more refined model than standard blockcyclic distributions to equally balance the load between processors of different speeds. We present several NP-completeness results that demonstrate the intrinsic difficulty of static load-balancing on heterogeneous dedicated platforms. We also show that column-based heterogeneous distributions are easy to build, efficient and give a unified framework. To cope with speed variations during computations, we propose a set of basic operations enabling to perform limited redistribution of data and computations while preserving column-based distributions.

Static schemes (to cope with heterogeneity) and remapping strategies (to cope with speed variations during computations) are theoretical tools that should make possible the design of a ClusterLAPACK.

3.9 Grid Application Development: The Ultimate Challenge for code portability and adaptivity

Lennart Johnsson
University of Houston and Royal Institute of Technology, Sweden

In a Grid environment, application codes must execute correctly on a variety of platforms. It is highly desirable that the efficiency of execution is good, ideally optimal, regardless of what data set and what platform a code is executed on. This goal forces a need for code adaptivity, i.e., that techniques be employed to select algorithms as well as code generation and scheduling techniques that adapt to both problems at hand and execution environments. In this presentation we will present an overall strategy towards a Grid Application Development Software system, and specific approaches and results for generating adaptive software for the Fast Fourier Transfrom, specifically the UHFFT software library package.

3.10 Thoughts about Data Management in Grid Environments

Andre Merzky
Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB)

Data intensive applications as Genom Sequenzing, High Energy Physics and Grand Challenge simulations act as strong driving forces for nowadays Grid development – the amount of data produced or processed by these applications seems not to be manageable by other techniques known today. Actually, Grid Data Management is also not yet existent, but it is a topic heavily worked on in various groups world wide. These efforts are coordinated by the 'Remote Data Access' working group of the 'Global Grid Forum', a standardization body for Grid middleware development.

The present talk describes the concept space of Grid Data Management as seen by this group, and exemplarily highlights recent advantages in some of its components: Data Transport, Storage System Properties, and Replica Management. The GGF working group proves its ability to drive actual development in this exciting area of large scale distributed computing.

3.11 Grid-aware Communication Components

Thilo Kielmann
Vrije Universiteit, Amsterdam

In Grid computing platforms, application-performance strongly depends on communication costs. Between the local and wide-area networks, the network performance differs by orders of magnitude. This asks for Grid-aware communication components. In this talk, I present three such components developed at our group. The first one is the Satin system for running parallel (Java) divide-and-conquer applications on Grid platforms. The second one is the Replicated Method Invocation mechanism (RepMI) that allows efficient sharing of Java objects. The third and last one is our MagPIe library that optimizes MPI's collective communication for Grid platforms.

The observation underlying Satin's system design is that divide-and-conquer parallelism is inherently hierarchical, matching the hierarchical LAN/WAN structure of Grid platforms – if the load-balancing takes the Grid cluster hierarchy into account. For Satin, we propose a new load balancing scheme, called cluster-aware random work stealing. This scheme is highly efficient, self-adaptive to (changing) WAN parameters, and trivial to implement.

Besides scheduling and load balancing, applications also need to share common data objects. For this purpose, we developed the Replicated Method Invocation scheme (RepMI) that resembles Java's RMI as much as possible while replicating objects to the participating processes. RepMI combines an efficient implementation with an easy-to-use programming interface. However, on Grid platforms, RepMI's consistency model may cause undesirably high communication overhead, especially when the application's locality of write operations is weak. Those applications can benefit from collective communication operations, like the ones known from MPI.

Our MagPIe library implements MPI's collective operations with minimal completion time for clustered Grid systems. MagPIe is based on the P-LogP performance model that covers the performance aspects needed for modelling collective operations. We are able to build collectives that efficiently use WAN links by using message segmentation and communication graph shape optimization. Work in progress is the integration of the various Grid-aware components in order to build a widely applicable Grid programming toolkit.

3.12 Knowledge Discovery and Data Mining in Grid-Based Distributed Environments

Domenico Talia, P. Trunfio
ISI - CNR, Rende

Knowledge discovery tools and techniques are used in an increasing number of scientific and commercial areas for the analysis of large data sources. When large data repositories are coupled with geographic distribution of data, users and systems, it is necessary to combine different technologies for implementing high-performance distributed knowledge discovery systems. On the other hand, computational grid is emerging as a very promising infrastructure for high-performance distributed computing. This talk presents a software architecture for parallel and distributed knowledge discovery (PDKD) systems built on top of computational grid services that provide dependable, consistent, and pervasive access to high-end computational resources. The architecture, named Knowledge Grid, uses the basic grid services and defines a set of additional layers to implement the services of distributed knowledge discovery process on grid-connected sequential or parallel computers.

This is joint work with Mario Cannataro and P. Trunfio.

3.13 The Internet2 Qbone Bandwidth Broker

Steve Chapin
Syracuse University, Dept. of EE & CS

The scheduling of computational resources in metacomputers has been well understood for several years, but the same cannot be said of the scheduling of networking resources. The standard practice has been to assume that adequate, appropriate networking resources will be available to serve any job. Bandwidth Brokers manage network resources within individual autonomous domains, and interact with peer bandwidth brokers to provide end-to-end Quality-of-Service guarantees to user processes. This talk describes the role, design, and use of Bandwith Brokers within the Internet2 Quality-of-Service Backbone (Qbone) to manage network bandwidth within IETF Differentiated Services (DiffServ) networks.

3.14 The NWIRE Resource Management Architecture

Ramin Yahyapour
Computer Engineering Institute, University Dortmund

This talk presents a management infrastructure for distributed resources in a Grid environment as developed in the research prototype during the NWIRE project. The work focuses on the specific requirements in job scheduling, as there is need for differentiated objectives, guarantees and reservation. As resources are provided by different owners and often not dedicated for grid use, site-autonomy has to been taken into account. Therefore the NWIRE infrastructure uses a domain-based approach that keeps the resource control fully to the local administration while an additional trader/broker (called MetaManager) interacts with other Grid domains.

Different scheduling implementations can interact with each other by using request and offer mechanisms. It is shown that market-economic approaches may be suitable for Grid scheduling. First simulations show results in the range of conventional scheduling methods as backfilling for common measures as completion time and utilization.

3.15 Integrating Local Resources into the Grid: The Resource Specification Problem

Jean-Marc Nicod
Université de Franche-Comté, Laboratoire LIFC

In this talk the architecture and the algorithms used in DIET (Distributed Interactive Engineering Toolbox) has been presented, a hierarchical set of components to build Network Enabled Server applications in a Grid environments. This environment is build on top of different tools which are able to locate an appropriate server depending of the request sent by a client, the data that may be computed in previous requests (and thus located on another server) and the dynamic performance characteristics of the system. The services developed to achieve the aim of DIET are based on the CORBA norme.

3.16 Integrating Local Resources into the Grid: The Resource Specification Problem

Alexander Reinefeld
Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB) and Humboldt Universität zu Berlin (HU)

In Grid computing environments, the execution of distributed applications requires the detection, the selection, and the (co-) allocation of resources. A resource, in this context, might be understood as a computer (e.g., a high-performance system or a simple PC), a network link (e.g., a system area network, a LAN- or WAN-link), a switch, a storage system (e.g., a tape robot), a special hardware device (e.g., a protein sequencing machine), a software package, or even a human expert service.

In this talk, we focus on methods for specifying resources in the Grid. We present a graph-based method that builds on the XML standard. It can be used for specifying both, resource offers and resource requests. Our method uses attributed hierarchical graphs, where the vertices represent processors (or processes) and the edges represent network links (or communication channels). Compared to the popular Globus approach, which uses the Resource Specification Language RSL for requests and the Metacomputer Directory Service MDS, our approach is symmetrical, that is, the same representation (and the same tools) can be used for both sides. Moreover, the matching between resource requests and resource offers can be done with existing graph matching algorithms. This is especially favorable for systems operated in space-sharing mode.

3.17 Monitoring and Performance Analysis in Distributed Systems

Roland Wismüller
Technische Universität München

Both the development of efficient Grid applications and the optimal management of Grid resources depend on the ability to monitor the run-time behavior of these applications and resources. The talk addresses two aspects: low level monitoring and high level performance analysis.

With the On-line Monitoring Interface Specification OMIS, a flexible and extensible interface between on-line tools and their target systems has been defined. Its basic concepts are an object-based model of the target system, the event/action-paradigm, services operating on sets of objects, and location transparency. The currently existing implementation for networks of workstations proves that monitoring requests can be executed in an efficient, distributed way. Thus, the OMIS interface and its implementation offer a good basis for Grid monitoring, too.

Performance analysis on the Grid imposes some specific problems. First, we have to deal with very large applications, where event tracing becomes impossible due to the amount of data that would have to be generated and transmitted. Second, the programmer usually doesn't know the architectural details of the Grid resources, so he can not easily evaluate the acquired data.

We thus think that the Grid requires an automated form of on-line performance analysis. Based on concepts developed in the APART working group, we propose a hierarchical agent based architecture for automatic performance evaluation and discuss different strategies for searching and locating performance bottlenecks.

3.18 How to Build a Reliable Cluster for the Grid

Florian Schintke
Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB)
Lord Hess
Kirchhoff-Institut für Physik, Universität Heidelberg

Two new paradigms are changing the way we do computing: Clusters and Grids. Both have been born by the need for more economical means for high-performance computing: Clusters employ cost-effective commodity components for building powerful computers, and Grids allow to better utilize the computing resources that are available on the Internet.

In this talk, we present the architectural framework of a HPC cluster that employs mass market components for high-performance computing. Our major design goals are scalability to some thousand nodes, reliability, and low maintenance overhead.

Building on the cluster as a compute node, we present middleware for the integration of compute servers in a worldwide grid environment. Here, the focus is on the management of distributed resources, work-load balancing, scheduling and other grid services.

4 List of Participants

Larry Carter
University of California at San Diego
Dept. of Computer Science & Engineering
9500 Gilman Drive
CA 92093-0114 La Jolla
phone: +1-858-534-62 65
fax: +1-858-822-1559
carter@cs.ucsd.edu
http://www.cse.ucsd.edu/users/carter/

Steve Chapin
Syracuse University
Dept. of EE & CS
NY 13244 Syracuse
phone: +1-315-443-4457
fax: +1-315-443-1122
chapin@ecs.syr.edu
http://www.hpdc.syr.edu/˜chapin

Dick H.J. Epema
Delft University of Technology
Faculty of Technical Mathematics and Informatics
Mekelweg 4
NL-2600 GA Delft
phone: +31-15-278-3853
fax: +31-15-278-7141
epema@cs.tudelft.nl
http://www.pds.twi.tudelft.nl/˜epema/

Jeanne Ferrante
University of California at San Diego
Dept. of Computer Science & Engineering
9500 Gilman Drive
CA 92093-0114 La Jolla
phone: +1-858-534-8406
fax: +1-858-822-1559
ferrante@cs.ucsd.edu
http://www.cse.ucsd.edu/users/ferrante/

Lord Manfred Hess
Universität Heidelberg
Kirchhoff-Institut für Physik
Schröderstr. 90
D-69120 Heidelberg
phone: +49-6221-54-4323
fax: +49-6221-54-4345
hess@kip.uni-heidelberg.de
http://www.kip.uni-heidelberg.de/

S. Lennart Johnsson
University of Houston
Dept. of Computer Science
592 Philipp G. Hoffman Hall
4800 Calhoun Street
TX 77204-3475 Houston
phone: +1-713-743-3371
fax: +1-713-743-3376
johnsson@cs.uh.edu
http://www.cs.uh.edu/~johnsson/

Thilo Kielmann
Vrije Universiteit Amsterdam
Wiskunde en Informatica
De Boelelaan 1081a
NL-1081 HV Amsterdam
phone: +31 20 44 47789
fax: +31 20 44 47653
thilo.kielmann@acm.org
http://www.cs.vu.nl/˜kielmann/

Arnaud Legrand
Ecole Normale Supérieure de Lyon
Laboratoire LIP
Projet CNRS-INRIA ReMaP
46 allee d'Italie
F-69364 Lyon Cedex 07
fax: +33-4-72 72 80 80
arnaud.legrand@ens-lyon.fr
http://www.ens-lyon.fr/˜alegrand/

Thomas Ludwig
Universität Heidelberg
Institut für Informatik
Im Neuenheimer Feld 368
D-69120 Heidelberg
phone: +49-6221-54-5750
fax: +49-6221-54-8877
thomas.ludwig@informatik.uni-heidelberg.de
http://www.informatik.uni-heidelberg.de/˜Thomas.Ludwig/

André Merzky
Konrad-Zuse-Zentrum für Informationstechnik
Dept. Visualization
Takustr. 7
D-14195 Berlin
phone: +49-30-841-85-339
fax: +49-30-841-85 107
merzky@zib.de
http://www.zib.de/merzky

Barton P. Miller
University of Wisconsin
Computer Sciences Dept.
1210 W. Dayton St.
WI 53706 Madison
phone: +1-608-263 3378
fax: +1-608-262 9777
bart@cs.wisc.edu
http://www.cs.uisc.edu/~bart/

Jean-Marc Nicod
Université de Franche-Comté
Laboratoire LIFC
30, Route de Gray
F-25030 Besançon
phone: +33-3-81 66 20 68
fax: +33-3-81 66 64 50
Jean-Marc.Nicod@univ-fcomte.fr
http://lifc.univ-fcomte.fr

Alexander Reinefeld
Konrad-Zuse-Zentrum für Informationstechnik
Computer Science Research
Takustr. 7
D-14195 Berlin
fax: +49 30 84185-311
reinefeld@zib.de
http://www.zib.de/reinefeld/

Volker Sander
Forschungszentrum Jülich GmbH
Zentralinstitut für Angewandte Mathematik
D-51425 Jülich
phone: +49-2461-61 65 86
fax: +49-2461-61 66 56
v.sander@fz-juelich.de

Florian Schintke
Konrad-Zuse-Zentrum für Informationstechnik
Computer Science Research
Takustr. 7
D-14195 Berlin
phone: +49-30-841-85-306
fax: +49-30-841-85-311
schintke@zib.de
http://www.zib.de/schintke/

Uwe Schwiegelshohn
Universität Dortmund
FB Elektro- und Informationstechik
Otto-Hahn-Str. 4
D-44221 Dortmund
phone: +49-231-755-26 34
fax: +49-231-755-32 51
uwe.schwiegelshohn@udo.edu
http://www-ds.e-technik.uni-dortmund.de/WEB-D/Mitarbeiter/schwiegelshohn.shtml

Achim Streit
Universität Paderborn
Center for Parallel Computing
Fürstenallee 11
D-33102 Paderborn
phone: +49-5251-60 63 31
fax: +49-5251-60 62 97
streit@uni-paderborn.de
http://www.uni-paderborn.de/pc2/

Domenico Talia
ISI - CNR
Via P. Bucci
I-87030 Rende
phone: +39-0984-831-725
fax: +39-0984-839-054
talia@si.deis.unical.it
http://isi-cnr.deis.unical.it:1080/˜talia/

Jon B. Weissman
University of Minnesota
Dept. of Computer Science & Engineering
Twin Cities Campus - 4-192 EE/CS
200 Union Street S.E.
MN 55455 Minneapolis
phone: +1-651-291-8898
fax: +1-612-625-0572
jon@cs.umn.edu
http://www.cs.umn.edu/~jon/

Roland Wismüller TU München
Institut für Informatik
LST Rechnertechnik & Rechnerorganisation
Arcisstr. 21
D-80290 München
phone: +49-89-289-28243
fax: +49-89-289-28232
wismuell@in.tum.de
http://wwwbode.in.tum.de/~wismuell/

Ramin Yahyapour
Universität Dortmund
FB Elektro- und Informationstechik
LST Datenverarbeitungssysteme
Otto-Hahn-Str. 4
D-44221 Dortmund
phone: +49-231-755-27 35
fax: +49-231-755-32 51
yahya@ds.e-technik.uni-dortmund.de
http://www-ds.e-technik.uni-dortmund.de/