1990

ATHENA: A Knowledge Base System for //ELLPACK

C. E. Houstis

Elias N. Houstis
Purdue University, enh@cs.purdue.edu

M. Katzouraki

T. S. Papatheodorou

John R. Rice
Purdue University, jrr@cs.purdue.edu

Report Number:
90-950

ATHENA: A KNOWLEDGE BASE
SYSTEM FOR //ELLPACK

C. E. Houstis
E. N. Houstis
M. Katzouraki
T. S. Papatheodorou
J. R. Rice
P. Varodoglou

CSD-TR-950
February 1990

ATHENA:
A Knowledge Base System for //ELLPACK*

C.E. Houstis, E.N. Houstis, M. Katzouraki,
T.S. Papatheodorou, J.R. Rice, and P. Varodoglou

Computer Sciences Department
Purdue University
Technical Report CSD-TR-950
CAPO Report CER-90-8
February, 1990

Abstract

We describe the design of a knowledge base and associated inference mechanism (ATHENA) for the expert system Parallel (//) ELLPACK. The objective of ATHENA is to provide combinations of grids, methods and machines which best meet a user's specified performance requirements for accuracy, ε, and response time, T. The knowledge base contains a large database of performance data and the inference mechanism is based on performance profiles derived from this database. The system can acquire new performance data silently during its usage and automatically update its performance profiles. It can also produce new performance data separately using either problems posed by "experts" or collected from users.

ATHENA: A KNOWLEDGE BASE SYSTEM FOR //ELLPACK

C.E. Houstis, E.N. Houstis, M. Katzouraki,
T.S. Papatheodorou, J.R. Rice and P. Varodoglou

ABSTRACT

We describe the design of a knowledge base and associated inference mechanism (ATHENA) for the expert system Parallel (//) ELLPACK. The objective of ATHENA is to provide combinations of grids, methods and machines which best meet a user's specified performance requirements for accuracy, ε, and response time, T. The knowledge base contains a large database of performance data and the inference mechanism is based on performance profiles derived from this database. The system can acquire new performance data silently during its usage and automatically update its performance profiles. It can also produce new performance data separately using either problems posed by "experts" or collected from users.

1. Introduction

We consider the design and development of a parallel knowledge base for the parallel (//) ELLPACK [Hous 89] system for solving certain types of partial differential equations (PDEs). Its design objective is to reduce the overhead associated with the parallel processing of these types of computations. Specifically, it will provide, a) facilities for the automatic partitioning and allocation of the PDE computations to a variety of parallel machines and b) expert assistance for selecting "efficient" method/machine pairs. The //ELLPACK system allows many alternative ways to solve elliptic PDEs so the selection of a good way becomes a nontrivial task. The ATHENA expert system is designed to be able to produce expert assistance for the *method/machine selection problem*. Recall that //ELLPACK is designed to run on a multilevel hardware facility consisting of powerful workstations, machines with hundreds of kMIPS processors (MIPs = millions of instructions per second) and machines with tens of BIPS processors (BIPS = billions of instructions per second). Thus the efficiency of the computation will depend critically on the machines. ATHENA's unique design is its use of performance profiles and its ability to generate new performance profiles, and thus better selection capabilities, as it is used. The database facility uses stochastic methods to rank methods and machines using the performance profiles, it also selects the most relevant profiles and evaluates their validity. New data can be incorporated during "training" runs as well as from normal use of //ELLPACK.

Section 2 describes the various performance evaluation data used by ATHENA. The software organization and design goals of ATHENA are described in Section 3. Finally, Section 4 describes the inference mechanism of ATHENA.

2. Performance Evaluation Data

It is clear that the main performance objectives of a user are *accuracy* and *time*. In a PDE computation *accuracy* is controlled through the refinement of the grid and the discretization scheme used, while *execution time* depends on the speed of the targeted machine and the efficiency of the PDE solver. For a given machine, the computation of a solution within a certain accuracy (ε) and time frame (T) requires the selection of appropriate grid, discretization and solution schemes (method) plus an appropriate machine. If the machine is parallel, then the partitioning of the computation into load balanced, optimally parallel subtasks is also required. For the various steps of the computation a number of performance indicators are measured. We present them as *performance profiles*, one for each combination of PDE problem, method and machine. See [Boisvert, Rice and Houstis, 1979], [Boisvert and Rice, 1985, Chapters 8-11 and Appendix A] for further details. For each such combination we collect, for different grid sizes, data on errors, execution time, linear system size, number of iterations and similar items. Aggregation techniques must be used in retrieving data, for no database can ever have all the data needed. The techniques used here include:

(a) *Machine Equivalences*. If we have execution time data for a VAX 11/780, a VAX 8800 and an Alliant FX80 then we use a conversion factor to estimate the execution time for all cases on any one of these three (or other) machines.

(b) *PDE Problem Associations*. For each PDE problem we collect (or can easily recompute) data on over 60 simple properties of the problem. Sixteen of these can be considered problem features (e.g., rectangular domain, no cross derivative term, Dirichlet boundary conditions) in the usual sense. In addition we have 36 possible features which are more subjective or computationally expensive (e.g., boundary layer present, variable smoothness, nearly singular). Eighteen of these refer to the PDE problem in general, eight to the operator and ten to the solution. These latter 36 features are graded on a scale of 0 to 100.

The machine equivalences allow us to extend the data to many machines in a straightforward manner. The PDE problem features allow us to take a new problem and find "close matches" to problems with existing performance data. Thus the aim of ATHENA is to collect several "close" problems and to estimate a performance profiles based on the existing data. The reliability of the estimate depends on the closeness and. of course, we must be prepared for the case where no relevant data exists, we discuss the action for this case later. In the case of parallel machines, the data points of the performance profiles are assumed to correspond to nearly optimal machine configurations for the corresponding grid sizes. Examples of such performance curves for sequential and parallel machines can be found in [Rice and Boisvert, 1985, Chapter 8-11], [Hous 88] and [Chri 88].

The ELLPACK project has accumulated an extensive database of performance measures for sequential machines, about 15,000 PDE solutions involving over 100 PDE problems, many methods and perhaps 10 combinations of compiler/operating system and machines. The data collected so far for parallel machines is, of course, still quite sparse.

3. ATHENA's Goals and Software Organization

ATHENA is an expert system whose knowledge base consists of performance profiles, which are automatically generated from a database of performance measurements and dynamically updated when the corresponding database is reorganized or enriched. The objective of ATHENA is to select the method (grid, discretization and solver) and machine based on the nature of the PDE problem and user's computational objectives (accuracy, time response). The software infrastructure of ATHENA consists of a performance evaluation facility, a facility for analyzing the data for various classes of problems, a facility for automatically generating performance profiles and an inference facility that provides an expert solution to selection problems.

The overall structure of the ATHENA system is shown in Figure 1. The //ELLPACK system [Houstis et. al., 1989] comprises the three boxes on the left while the ATHENA system is on the right.

Parallel Machines

If the selection process involves a parallel machine, then there is a difficult subproblem which the //ELLPACK system solves, namely selecting the number of processors, decomposing the PDE problem into parts and assigning these parts to the processors. This subproblem logically follows that of selecting a discretization and grid pair (as the accuracy requirement determines these). Once such a pair is identified, one can estimate the time and memory requirements roughly for various parallel machines and make a machine selection. Once this is done we have a method and a machine, but we must specify in detail the mapping of the PDE problem onto the parallel machine. This final step is carried by the //ELLPACK system.

3.1. Performance evaluation and knowledge acquisition facility

Figure 2 shows a block view of the performance database, its data acquisition facility and its data analyzer or performance estimate generator. The primary objective of this facility is to carry out experiments to enrich the performance database. This facility is also used to collect automatically data from the user's //ELLPACK program and dynamically update the database. A knowledge base manager keeps track of these operations. The //ELLPACK system is at the center of this facility as also seen in Figure 1. The performance evaluation generated from the database here is to guide the experimenter rather than to help the user select a method but the computations and software are very similar.

The data analyzer is for locating data relevant to the PDE problem at hand. The PDE problem associations are made based on a comparison of the vector of problem properties and features of both the new PDE problem and those in the database. A metric is used to measure the "distances" and one of the key steps in the construction of this facility is to make this measure reliable. The properties and features have both logical and numerical values so the metric has components which are numerical weights

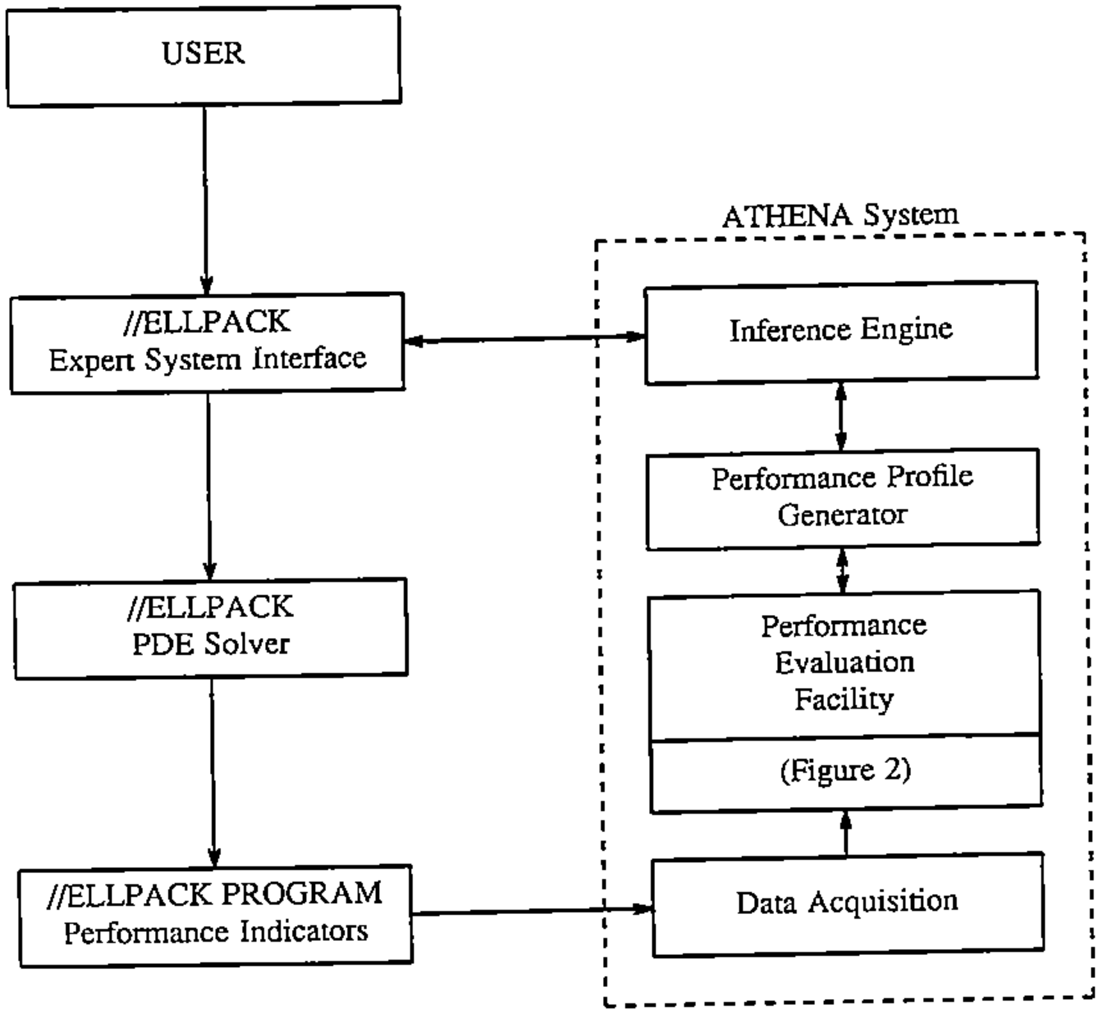

Figure 1: The organization of ATHENA's software structure.

associated with these values. The machine equivalences are also applied by this analyzer.

Note that this facility can also be used to tune the inference engine and data analyzer. An ''expert'' may observe the results of the inference engine (playing the role of a user) and then use this facility to review the intermediate steps in the generation of a selection. He may also experiment with changing various weights and values in these processes.

3.2. Knowledge generation

The primary representation of the knowledge are the performance profiles which relate time and accuracy to machines, discretization, solvers, and grids. These profiles depend on the PDE problems and so the larger the set of problems, the better chance of finding profiles relevant to a particular user's PDE. The ATHENA system attempts to

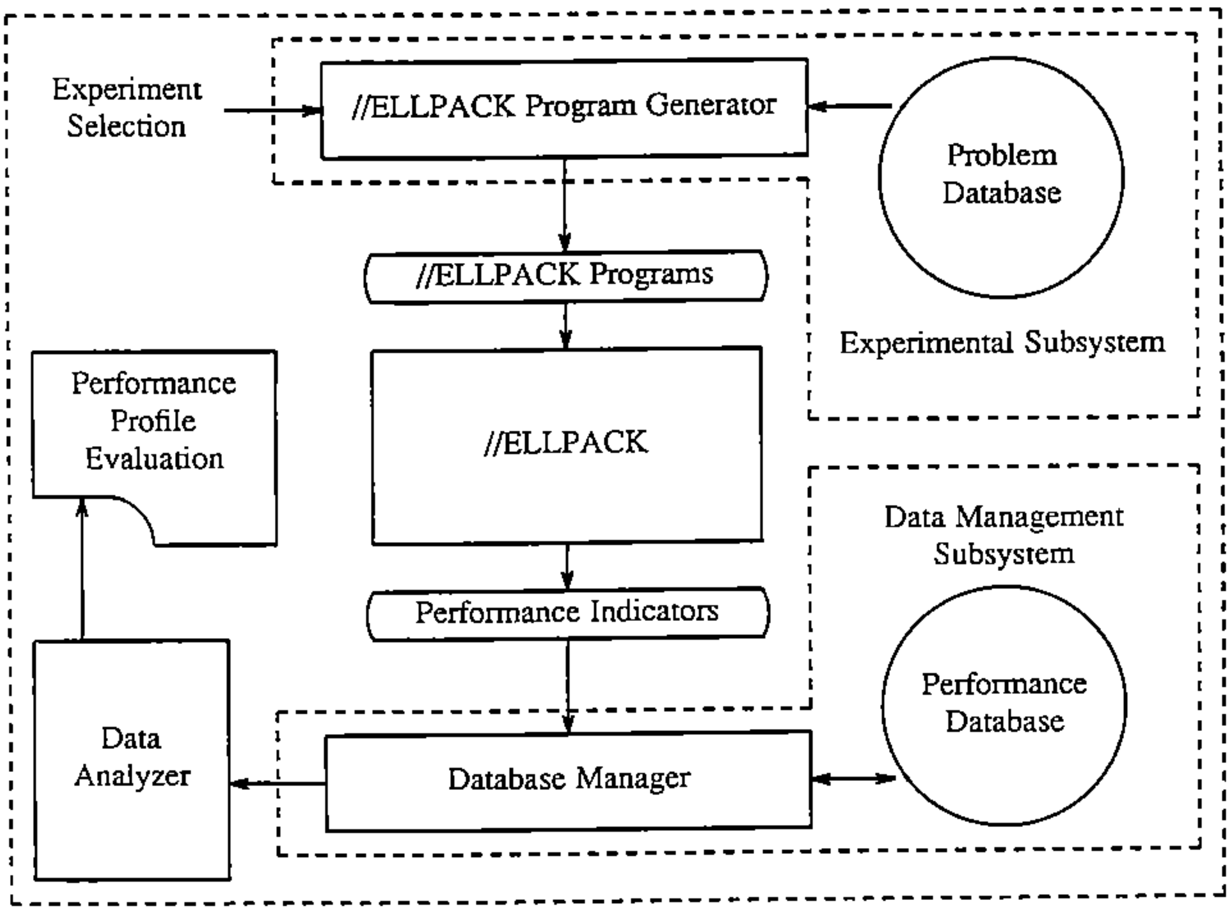

Figure 2: Structure of the performance evaluation and knowledge acquisition facility for ATHENA.

synthesize performance profiles from data about problems that are "close" to the user's PDE. As the database grows, cluster analysis techniques are used to identify new problems (perhaps completely artificial ones) for which there is sufficient data to generate a reliable set of performance profiles. Once such a situation is identified, the new problem is "created" in the database and the synthesized profiles computed and made available for later use. In this way the use and training of //ELLPACK allows the ATHENA system to generate new knowledge about how to select methods and machines.

3.3. Performance estimation

The ATHENA system extracts information about the PDE problem from the user interface where the problem is formulated. It also asks for information, especially about features, from the user. Using this information, PDEs in the knowledge base are located which are close to the given one. The data analyzer then evaluates the relevance and closeness of the problems located and synthesizes performance profiles for this problem on "standard" sequential and parallel machines. These standard machines can be equivalenced to any machines actually available for solving the PDE. This process is illustrated in Figure 3.

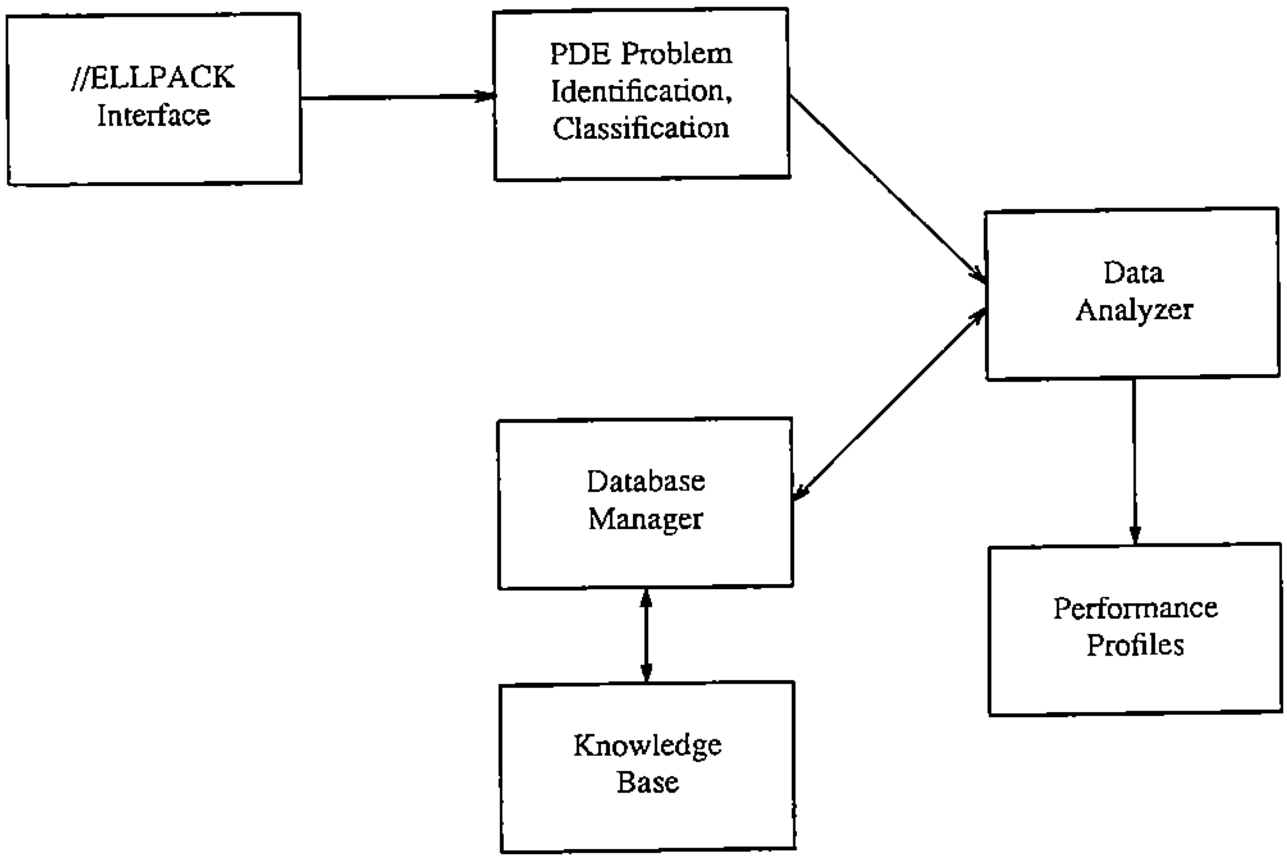

Figure 3. Schematic of ATHENA's use of the knowledge base to synthesize performance profiles upon which performance estimates are based.

3.4. "New" PDE problems

The ATHENA system must be prepared for PDE problems where little or no relevant information can be located in the knowledgebase. The problem might be one truly different from any seen before or it might be one where most of the features are unknown and not readily computed. The system, of course, asks for guidance from the user when features are missing, but the user may choose not to respond in a helpful way. In such a case, the system requests permission to make exploratory computations. If so permitted, it chooses a very robust, general discretization (say, collocation with Hermite cubics), a coarse mesh (say, 7 by 7), a robust solver (say, Gauss elimination), and a convenient machines (say, the user's workstation) and solves the PDE. The mesh is refined a little and the data collected is examined to see if systematic performance behavior is present. The results are also displayed to the user in an attempt to prod him into giving guidance. However, if no further guidance is given, the system continues along a predetermined path until either the problem is solved or the time response limit is reached.

4. ATHENA's Inference Engine

The purpose of the inference engine is to make selections of method and machines using the performance profiles synthesized for the PDE problem at hand. It also uses rules (in the usual sense of rule based expert systems) that serve to focus the inferences and to resolve uncertainties in the selection. The inference starts with the PDE already classified and a certain number of performance profiles available. The steps in the inference are as follows:

1. Identify all applicable methods.
2. Eliminate methods which.
   a) are generally inferior to other applicable methods
   b) have no performance profiles
3. For each method (discretization) use the performance profiles of accuracy versus grid to estimate grid size required.
4. For each discretization method use the performance profiles to estimate the execution time for solvers (on the "standard" machines). Eliminate grossly inferior solvers.
5. Convert execution time estimates from standard machines to available machines.
6. Query available machines about estimated response for computations that are likely to meet the time requirements.
7. Select the method and machine which meets the requirements and, if possible, meets auxiliary objectives such as: Lowest cost, small impact on network, most confidence in estimates, etc.

These steps are shown in schematic form in Figure 4.

Note that parts of this process is similar to that in *Elliptic Expert* [Dyksen and Gritter, 1989]. The principal enhancements of ATHENA are, a) the heavy reliance on performance profiles, b) the synthesizing of new performance profiles, c) the inclusion of parallel machines in the selection, d) the enhancement of the knowledge base during use of //ELLPACK, and e) the experiment system for "training" ATHENA.

References

[Bois 79] Boisvert, R.F., E.N. Houstis and J.R. Rice, A system for performance evaluation of partial differential equation software", *IEEE Trans. Software Engr.*, 5 (1979), pp. 418-425.

[Dyks 89] Dyksen, W.R. and C.R. Gritter, Eliptic Expert: an expert system for elliptic partial differential equations, *Math. Comp. Simulation*, **31** (1989), pp. 333-342.

[Hous 88] Houstis, E.N., J.R. Rice, C.C. Christara and E.A. Vavalis, Performance of scientific software, *Mathematical Aspects of Scientific Software* (ed., J.R. Rice), IMA Volumes in Mathematics and its Applications **14**, Springer-Verlag, New York (1988), pp. 123-155.

[Hous 89a] Houstis, E.N., J.R. Rice and T.S. Papatheodorou, Parallel (//) ELLPACK: An expert system for the parallel processing of partial differential equations, *Math. Comp. Simulation* **31** (1989), pp. 497-508.

[Hous 89b] Houstis, E.N., M. Katzouraki, T.S. Papatheodorou and V. Sotiropoudou, Logic parallelism in an expert system for solving partial differential equations, *Intelligent Mathematical Software Systems*, (eds., E.N. Houstis, J.R. Rice and B. Vichnevetsky) Elsevier, Amsterdam (1990), to appear.

[Rice 85] Rice, J.R. and R.F. Boisvert, *Solving Ellpitic Problems Using ELLPACK*, Springer-Verlag, New York (1985).