

Parameter Scan of an Effective Group Difference Pseudopotential Using Grid Computing

Wibke SUDHOLT and Kim K. BALDRIDGE

*Department of Chemistry & Biochemistry and San Diego
Supercomputer Center, University of California, San Diego,
9500 Gilman Drive, Mail Code 0505, La Jolla, CA 92093-0505, USA*

David ABRAMSON, Colin ENTICOTT and Slavisa GARIC

*School of Computer Science and Software Engineering,
Monash University, Clayton, Victoria, 3800 Australia*

Received 26 June 2003, revised 2 December 2003

Abstract Computational modeling in the health sciences is still very challenging and much of the success has been despite the difficulties involved in integrating all of the technologies, software, and other tools necessary to answer complex questions. Very large-scale problems are open to questions of spatio-temporal scale, and whether physico-chemical complexity is matched by biological complexity. For example, for many reasons, many large-scale biomedical computations today still tend to use rather simplified physics/chemistry compared with the state of knowledge of the actual biology/biochemistry. The ability to invoke modern grid technologies offers the ability to create new paradigms for computing, enabling access of resources which facilitate spanning the biological scale.

Keywords Grid Computing, Distributed Parametric Modeling, QM/MM Methods, Pseudopotentials.

§1 Introduction

Advances in grid technology promise to offer novel modes of coupling scientific models and unique strategies of sharing and federating data, which, in the life sciences, can lead to the ability to bridge the gaps in our knowledge of biological complexity. Detailed understanding of structure/function relationships and

molecular reaction processes in complex biological systems can leverage sophisticated computer-based information handling tools, and new high throughput technologies thereby enabling levels of information content which push research developments to new heights.

In this work, we illustrate a new conceptual approach for computational investigations that involve many steps of processing, bookkeeping, and a need for substantial repetitive computation over several variant parameters. The work involves the coupling of the GAMESS quantum chemical code to the Nimrod/G grid distribution tool within the PRAGMA project, to demonstrate the utility of grid infrastructure to significantly reduce the work involved in this parameterization procedure.. The technology demonstrated here significantly extends the manageability of accurate, but costly quantum chemical calculations and is thus valuable for a wide range of computational life sciences studies, beyond what is demonstrated here.

§2 Motivating Application

A large component of biomedical research involves numerical experimentation and hypothesis testing. Searching parameter space for optimal solutions is a key area in which computational requirements are amplified by orders of magnitude. An important application of theoretical modeling is the simulation of extended molecular systems such as solutions, materials, and biomolecules. This is challenging because of their large sizes and the required sophisticated methodologies. Fortunately, many such systems can be partitioned into a small “active” region, which needs to be treated accurately, and a surrounding larger, “inactive” part, which can be modeled more approximately. This is the concept behind hybrid quantum mechanics-molecular mechanics (QM/MM) techniques¹⁾ (see Fig. 1(a)), where the “active” region is described based on the Schrödinger equation, while the “inactive” region is treated using classical force fields. However, since the physics of both methodologies is quite different, their coupling is difficult. One area of concern is the cutting of chemical bonds between the two parts. Whereas in the classical region dangling bonds can simply be eliminated, the outermost atoms of the quantum region would become unrealistic radicals.

Zhang et al.²⁾ recently introduced the “pseudobond” approach to saturate these atoms: The first atom of the MM region is included as a capping atom in the QM calculation, but parameterized such that it reflects the properties of the cut bond (see Fig. 1(b)). Therefore, they modified a fluorine atom using

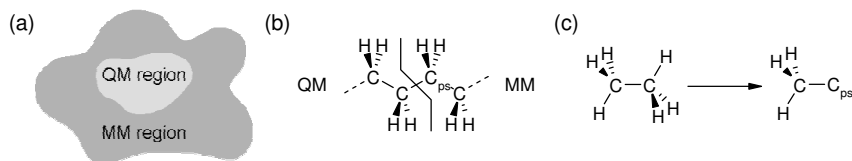


Fig. 1 (a) Concept of the QM/MM approach. (b) Partitioning of a system with the pseudobond method. (c) Parameterization with ethane ($C_{ps} = F$ with ECP or GDP).

an effective core potential (ECP) to mimic the isoelectronic methyl group in a carbon-carbon single bond situation (see Fig. 1(c)). Their parameterizations were first done for the ethane molecule, then tested on ethane derivatives, and subsequently applied for investigations of enzymatic reactions using the derived QM/MM techniques. Other, similar approaches also exist in the literature. Unfortunately, during exploratory studies we came across serious drawbacks of the Zhang et al. method: Calculations break down under certain conditions and pseudoatom ECPs appear rather difficult to optimize in general.

In the present study, we approach this question from a different perspective. We have developed a new formulation for an effective pseudoatom potential, which only deals with the discrepancies between methyl and fluoro groups and is thus named “group difference potential” (GDP). Its functional form is basically the superposition of two unmodified Gaussian functions,

$$U_{\text{eff}}(r) = A_1 \exp(-B_1 r^2) + A_2 \exp(-B_2 r^2). \quad (1)$$

Evaluation of the corresponding molecular properties with GAMESS³⁾ then dictates a set of parameter values for A_1 , A_2 , B_1 , B_2 that minimizes a cost function value, in this case of the normalized least square differences expression

$$f(A_1, A_2, B_1, B_2) = \frac{1}{\sum_{i=1}^{32} w_i} \sum_{i=1}^{32} w_i \left(\frac{x_i - X_i}{u_i} \right)^2. \quad (2)$$

Here, w_i are “weighting” factors which account for the occurrences of each property x_i so that carbon and hydrogen features are of equal importance. The u_i are “unifying” factors, which correct for the apparent, or desired, accuracy of each property and are chosen from chemical intuition. Since the analysis of equation (2) is carried out after completion of each tuple of GAMESS jobs, the w_i and u_i values can be easily adapted to enable smooth minimization of f in forthcoming parameter optimizations.

To avoid trapping in local minima, a portion of the parameter space is scanned in its entirety, which we have exemplified for pseudoethane. This task consists of thousands of short, uncoupled QM calculations and hence is a perfect grid computing application. The identification of low cost regions of the resulting cost function hypersurface will later facilitate the straightforward optimization of such pseudopotentials. Another intention in this work is to understand the effects of functional groups in the case of CH_3 and F. These chemical substitutions play an important role in life science investigations from synthetic organic chemistry to pharmaceutical drug development. In a broader context, this exercise demonstrates how grid infrastructure can be utilized to manage large multi-dimensional parameter sweeps and subsequent analysis using computational (bio)chemistry software with a variety of potential applications.

Such a formulation implies running GAMESS repetitively over a cross product of all values under consideration, some 15,000 independent jobs in our example. Performing this by hand on a single machine would be laborious, and manually on a distributed computational grid, almost impossible as well as error prone. However, by invoking the tool Nimrod/G⁴⁾, which has been specifically designed to perform parameter sweeps using resources distributed across a wide area computational Grid⁵⁾, the problem becomes tractable. Nimrod/G manages the experiment by finding suitable machines, sending input files to them, running a computation (which in this case involved the GAMESS package), and shipping the output files back to a central machine. More importantly, the software handles events such as network and node failures, the latter of which is a common occurrence over a large computational grid.

The original version of Nimrod was developed in 1994, and targeted only workstation clusters. The current version, called Nimrod/G, targets very wide area networks as characterized by the Global Grid. Fig. 2 shows the architecture of the Nimrod/G system. At the core of the system is a database which stores all of the details of one or more experiments. For example, the database contains entries about the individual jobs in a parameter sweep, whether they are currently allocated to resources, and if so, which resources. The database is loaded by two tools called the *Creator*, and the *Generator*, and is managed by conventional database server technology, in our case, PostGres.

Jobs are scheduled by a tool called the *Job Scheduler*. This application considers the various constraints, such as a soft real time deadline for the experiment and the cost of various resources, and notionally allocates jobs to resources.

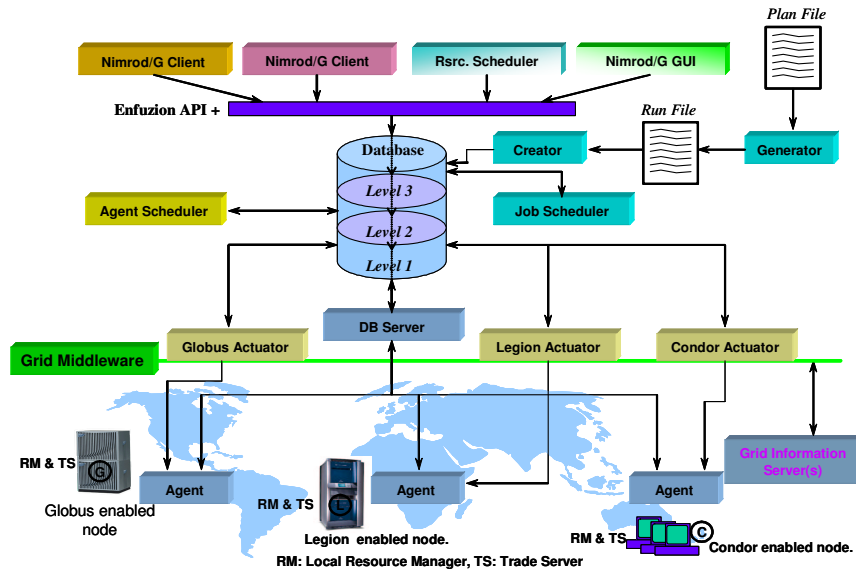


Fig. 2 Overall Nimrod/G Architecture.

Different computational resources on the Grid will have different costs associated with them, so Nimrod/G has provided a way performing experiments taking into account the user specifying time and/or cost constraints. The user may specify, “please complete this experiment using no more than 8,000 cost units”. This will prompt Nimrod/G to launch the experiment on each resource, when the cost of the resource can be determined (cost per hour vs. computational power), Nimrod/G will then estimate which resources to use to complete the experiment within budget and as quickly as possible. On a time requirement, Nimrod/G will behave similarly, except it will ease off on the more expensive resources if the experiment can be completed in time.

Jobs are actually run by a software component called the *Agent*. *Agents* are themselves scheduled to run on various resources. Once they start execution, *Agents* contact the database and request jobs, which are run sequentially on the resource. If a particular machine has more than one processor, then multiple *Agents* are scheduled for execution. This architecture helps hide the latency involved in scheduling and invoking a computation on a remote machines because each *Agent* can run more than one job. This is similar to the “Glide-in” mechanism used by Condor⁶). Because the Grid resources are heterogeneous, Nimrod must determine the type of architecture before it can run an agent. Once this

is done, it copies the *Agent* binary for the correct machine to the resource and starts the *Agent* running. The architecture information is extracted from the Globus MDS, along with information about the number of processors. Using this information Nimrod determines how many *Agents* to start.

Nimrod/G is built on a variety of middleware layers, including Condor, Legion and Globus. Globus, which is by far the most widely deployed toolkit, provides a uniform interface to the testbed resources regardless of their architecture, configuration or operating system⁷. As a result, Nimrod does not need to concern itself with the type of scheduler that is actually installed on a resource — it simply calls the GRAM (Globus Resource Allocation Module) interface, and Globus maps this to the underlying scheduler, such as PBS, NQE, Grid Engine, etc. Globus also provides a PKI security layer, and thus Nimrod users only need to obtain a valid certificate in order to use a resource.

In order to create a computational experiment for use by Nimrod/G, a user builds a short “plan” file. This file contains a textual description of the parameters under consideration along with the commands required to run the code. This set of commands is used by the *Agent* component. *Agent* interprets a task it receives and executes all of the commands, which with a single parameter set runs an experiment job.

The user needs to choose a set of suitable resources. Because GAMESS³ is available on so many platforms, this is not difficult. For this experiment, a testbed was built containing conventional workstations, clusters and vector supercomputers. As seen from Table 1, resources spanned a range of organizations, administrative domains, queue managers, countries, operating systems and architectures. In practice, some of the machines did not actually perform any work in this particular experiment, either due to their work load, or some problem associated with the software configuration. The highly dynamic nature of the Grid means that the decision about which machines are used on any particular experiment is deferred until the time that it is actually performed, and need not be made a priori.

There are currently a number of implementations of the Nimrod/G user interface. The one we used for this work was the Nimrod/G Portal, a web site that enables a user to create a plan and manage a computational experiment through a conventional browser. One of the most significant advantages of the portal is that it is not necessary to port the Nimrod/G client to the machine on which the user wants to launch the experiment, all of which is instead controlled

Table 1 Actual hardware used in the described experiment.

Machine name	Queue manager	Processors available	OS & architecture	Location
hathor.csse.monash.edu.au	PBS	24	Linux x86	Melbourne
brecca-2.vpac.org	PBS	30	Linux x86	Melbourne
koume.hpcc.jp	Grid Engine	4	Linux x86	Japan
ume.hpcc.jp	Grid Engine	64	Linux x86	Japan
amata1.cpe.ku.ac.th	SQMS	15	Linux x86	Thailand
erikson.ucsd.edu	PBS	74	Linux x86	San Diego
slic00.sdsc.edu	PBS	148	Linux x86	San Diego
hpc420.hpcc.jp	PBS	14	Solaris Sparc	Japan
tardis.eng.monash.edu.au	NQE	1	Cray SV1	Melbourne
sn9280.cray.co.jp	NQE	1	Cray SV1	Japan
venus.gridcenter.or.kr	PBS	64	Linux x86	Korea
jupiter.gridcenter.or.kr	PBS	16	Linux x86	Korea
apbs.rocksclusters.org	PBS	12	Linux x86	San Diego
chemcca40.ucsd.edu	PBS	36	Linux x86	San Diego

from a central location through one web site. The Nimrod Portal manages issues related to the underlying Grid middleware, such as the names of resources and Globus certificates. The portal also provides data about resources if they can not be located in the Globus MDS, such as the type of architecture. The portal allows the user to specify the jobmanager type (queue or fork) and the number of agents to execute, which removes the need for the MDS.

Once all of the jobs have completed and the output files are returned to the user, it is necessary to collapse the results into a form that can be interpreted. We chose to use the scientific visualization package, OpenDX (<http://www.opendx.org/>), to display the cost function value. However, because there are four input parameters, we needed to produce a sequence of visualizations each showing isosurfaces of cost value across three of the input parameters and a different frame for each of the fourth parameters. When concatenated into a movie, such displays allow us to explore the entire surface across all four parameters.

§3 Results and Discussion

Grid resource utilization during our first production experiment lasted some 42 hours, using a variety of remote resources. We have tools that graphically show the number of jobs running at any instant. Such graphs convey one of the more interesting aspects of the Grid, namely the ability to dynamically adjust which resource provides a particular service. For example, the largest number of jobs was executed by the cluster at HPCC in Japan. This was because that particular machine had the most processors of any other resource that could be dynamically assigned to the experiment. At the other extreme,

Table 2 Resource job statistics.

Resource	# of CPUs peak	# of CPUs reported by MDS	Total number of jobs	Total days	Total exe- cution time	Average job exe- cution time
brecca-2.vpac.org	29	186	4648	19 days	21:11:08	0:06:10
apbs.rocksclusters.org	2	12	143		15:05:57	0:06:20
slic00.sdsc.edu	7	148	443	2 days	21:08:46	0:09:22
erikson.ucsd.edu	32	76	3965	35 days	09:12:01	0:12:51
hathor.csse.monash.edu.au	36	36	2523	48 days	20:45:40	0:27:53
ume.hpcc.jp	57	64	3178	69 days	04:05:33	0:31:21
koume.hpcc.jp	4	8	255	5 days	21:37:46	0:33:19
chemcca40.ucsd.edu	13	36	721	18 days	00:17:15	0:35:58
		566	15876	200 days	17:24:06	0:18:12

the APBS ROCKs cluster in San Diego only ran a few jobs, presumably because it did not have spare capacity at that time. Nimrod/G actually incorporates scheduling heuristics that enable load movement in order to meet soft deadlines, a feature which leverages the dynamic properties of the Grid. *Most relevant to this experiment however was that we were not able to accumulate the number of processors required to complete this work within 42 hours at any one of the sites.*

Table 2 shows the amount of work done by each computational resource, but needs to be evaluated carefully. For example, even though ume.hpcc.jp had the highest number of executing jobs and provided the most execution time, because the machine did not have the fastest processors, it did not produce the largest number of results. In fact, brecca-2.vpac.org was able to execute the jobs faster and thus was able to produce more results with fewer CPUs. Overall, we were able to execute over 200 days of processing in about 42 hours.

Even though the experiment was very successful, we did experience a number of problems in setting up this large Grid testbed. With exception of network related and individual server problems, the biggest problem was associated with misconfigured Globus installations. Nimrod itself does not need to be installed on the remote resources, but it requires two Globus services, MDS and GRAM. Most of the resources we used had GRAM installed correctly, but did not have the MDS installed or configured with the jobmanager information that was set up for GRAM. Further, a bug in Globus' MDS PBS information (fixed in version 2.4.3) resulted in no diagnostic as to the total number of nodes or the total number of free nodes. One misconfigured resource referred to itself as localhost rather than the fully qualified domain name, causing Nimrod/G to fail while trying to gather information about that resource. The Globus developers will be addressing these types of problems in later versions, but in the meantime

we have implemented a temporary solution that disables jobmanagers while they are not needed. An additional problem experienced with Globus caused a few resources to go down due to the jobmanagers creating a high load on the resources' front-end. In the future, matters involving the installation and configuration of Globus should take care of such problems.

§4 Conclusions

The paradigm shift in biology towards a research and discovery process that is increasingly information-driven is enabling computational studies that are more tightly coupled to experimental studies. The rapid growth of grid technologies facilitates the coupling of key software, data and analysis tools, and the development of first-generation grid-enabled biology and chemistry software for complex research in important areas of health and disease. Linking together sophisticated methodologies in the manner exemplified here enables new integration pathways to discovery which can be carried out, automated, and repetitively performed with variant parameters, constraints or input datasets. Additionally, essential end-to-end audit of the whole process is an implicit deliverable.

In collaboration with several international groups, as part of the Pacific Rim Applications and Grid Middleware Assembly (PRAGMA), the project highlighted here illustrates access to global resources and application technologies that have been, and which will be further developed via simple web interfaces.

In subsequent studies, extensions to automate refinement of parameter space and parameter selections which drive automatic minimization runs, are planned. The Nimrod/O tool, a variant of Nimrod/G that performs automatic optimization⁴⁾, will be incorporated to invoke a number of search heuristics (e.g., gradient descent, simplex, evolutionary algorithms), enabling more intelligent search capabilities and advanced control of the search process.

Overall, the hybrid technology considerably reduces the development time of GDPs for applications involving organic molecules or functional groups. The more than $15,000 \times 4$ uncoupled QM calculations are systematically generated for a multidimensional grid of points, optimally distributed over several remote computing clusters, within a few days. The completion of such a number of runs would not have been possible in a reasonable timeframe without such technology. Results allow users a much better conceptualization of the parameter optimizations, thereby providing more insight into the physics behind the described phenomenon.

Acknowledgment We thank PRAGMA and all organizations which provided computers and the system administration, in particular the SDSC ROCKs group. We are grateful to J. P. Greenberg and K. Thompson, SDSC, for aiding software installations. W. S. acknowledges support from J. A. McCammon, UCSD, and the Postdoc Fellowship Program of the German Academic Exchange Service (DAAD). Cluster facilities were also sponsored by the National Biomedical Computational Resource (NBCR) and the W. M. Keck Foundation. K. B. acknowledges support from the NSF through DBI-0078296 and ANI-0223043 and from the NIH through NBCR-RR08605. We thank D. Kurniawan, Monash University, for visualization support. The Nimrod project is supported by DSTC and GrangeNet, both funded in part by the Australian Government. D. A. acknowledges support from the Australian Partnership for Advanced Computing (APAC) while on leave at UCSD.

References

- 1) Gao, J. and Thompson, M. A., (eds.), *Combined Quantum Mechanical and Molecular Mechanical Methods*, American Chemical Society, Washington, 1998.
- 2) Zhang, Y., Lee, T.-S. and Yang, W., "A Pseudobond Approach to Combining Quantum Mechanical and Molecular Mechanical Methods", *J. Chem. Phys.* *110*, pp. 46–54, 1999, and subsequent articles.
- 3) Schmidt, M. W., Baldrige, K. K., Boatz, J. A., Elbert, S. T., Gordon, M. S., Jensen, J. H., Koseki, S., Matsunaga, N., Nguyen, K. A., Su, S. J., Windus, T. L., Dupuis, M. and Montgomery, J. A., *J. Comput. Chem.* *14*, pp. 1347–1363, 1993; <http://www.msg.ameslab.gov/GAMESS/GAMESS.html>.
- 4) Abramson, D., Sobic, R., Giddy, J. and Hall, B., "Nimrod: A Tool for Performing Parametised Simulations Using Distributed Workstations", in *The 4th IEEE Symposium on High Performance Distributed Computing*, Virginia, August 1995; Abramson, D., Giddy, J. and Kotler, L., "High Performance Parametric Modeling with Nimrod/G: Killer Application for the Global Grid?", in *International Parallel and Distributed Processing Symposium (IPDPS)*, Cancun, Mexico, May 2000; <http://www.csse.monash.edu.au/~david/nimrod/>.
- 5) Foster, I. and Kesselman, C. (eds.), *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann Publishers, USA, 1999.
- 6) Frey, J., Tannenbaum, T., Foster, I., Livny, M. and Tuecke, S., "Condor-G: A Computation Management Agent for Multi-Institutional Grids", in *Proceedings of the Tenth IEEE Symposium on High Performance Distributed Computing (HPDC10)*, San Francisco, California, August 2001.
- 7) Foster, I. and Kesselman, C., "Globus: A Metacomputing Infrastructure Toolkit", *Int. J. Supercomput. Appl.* *11*, pp. 115–128, 1997.