Available online at www.sciencedirect.com
SCIENCE DIRECT®
Parallel Computing 30 (2004) 1057–1071
PARALLEL COMPUTING
www.elsevier.com/locate/parco

Evolutionary molecular structure determination using grid-enabled data mining

Mark L. Green, Russ Miller *

Department of Computer Science and Engineering, Center for Computational Research, State University of New York, Buffalo, NY 14260, USA

Received 20 April 2004; revised 1 July 2004; accepted 15 July 2004

Abstract

A new computational framework is developed for the evolutionary determination of molecular crystal structures using the *Shake-and-Bake* methodology. Genetic algorithms are performed on the *SnB* results of known structures in order to optimize critical parameters of the *SnB* computer program. The determination of efficient *SnB* input parameters can significantly reduce the time required to solve unknown molecular structures. Further, the grid-enabled data mining approach that we introduce exploits computational cycles that would otherwise go unused.

Keywords: Structure determination; Grid computing; Data mining; Genetic algorithms; Shake-and-Bake; Evolutionary methods

1. Introduction

The ACDC-Grid [1–5] is a proof-of-concept grid that has been implemented in Western New York. The driving application provides a cost-effective solution to the problem of determining molecular structures from X-ray crystallographic data

* Corresponding author.
E-mail addresses: mlgreen@ccr.buffalo.edu (M.L. Green), miller@buffalo.edu (R. Miller).

0167-8191/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.parco.2004.07.011

via the *Shake-and-Bake* direct methods procedure. *SnB* [6], a computer program based on the *Shake-and-Bake* method [7,8], is the program of choice for solving such structures in numerous laboratories [9–11]. This computationally-intensive procedure can exploit the grid's ability to present the user with a computational infrastructure that will allow for the processing of a large number of related molecular trial structures [12,13].

SnB has been used in a routine fashion to solve difficult atomic resolution structures, containing as many as 1000 unique non-Hydrogen atoms, which could not be solved by traditional reciprocal-space routines. Recently, the *Shake-and-Bake* research team has extended the application of *SnB* to solve heavy-atom and anomalous-scattering substructures of much larger proteins, provided that 3–4 Å diffraction data can be measured. In fact, while direct methods had been applied successfully to substructures containing on the order of a dozen selenium sites, *SnB* has been used to determine as many as 180 selenium sites. Such solutions have led to the determination of complete structures containing hundreds of thousands of atoms.

As shown in Fig. 1, X-ray data and the corresponding molecular structure are related by a Fourier transform. The X-ray data that is collected typically consists of positions and intensities. However, the corresponding phases are lost in the data collection process. Given positions, intensities, and phases, a Fourier transform can be used to generate the positions of the atoms in the molecule. Therefore, the solution to the problem of determining structures from X-ray data is referred to as *the phase problem*. That is, once the phases are determined, the positions of the atoms for the structure in question are easily derived.

The *Shake-and-Bake* procedure consists of generating structure invariants and coordinates for a (large) set of randomly generated "trial" structures. Each such trial structure is subjected to an automated cyclical procedure (refer to Fig. 2) between real space (where atoms live) and reciprocal space (where phases live). The *Shake-and-Bake* procedure includes (a) computing a Fourier Transform to determine phase values from the existing set of atoms, (b) determining a figure-of-merit [14] associated with these phases, (c) refining these phases locally against this figure-of-merit, (d) computing a Fourier Transform to produce an electron density map, and (e) employing a peak-picking routine to examine the map and find the maxima. These peaks (maxima) are then considered to be atoms, and the cyclical process is repeated for a user-determined number of cycles.

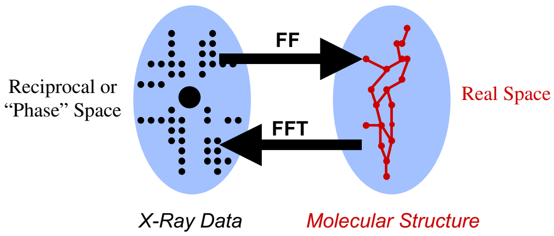


Fig. 1. X-ray data and corresponding molecular structure.

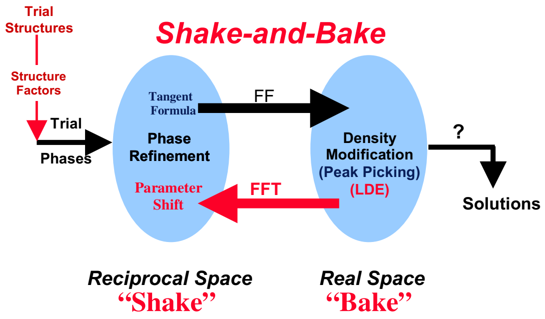


Fig. 2. *Shake-and-Bake* cyclic peak-picking procedure.

The procedure has a philosophy of "run until solved" in that one trial structure after another is subjected to the cyclical procedure until the figure-of-merit determines that one of these initially random trial structures has morphed into the correct structure. In practice, the random trial structures are generated on the fly, either sequentially or in parallel, as is appropriate for the computing platform that is being used.

The running time of *SnB* varies widely as a function of the size of the structure, the quality of the data, the space group, and choices of critical input parameters, including the size of the Fourier grid, the number of reflections, the number and type of invariants, the number of cycles of the procedure used per trial structure, and critical real-space and reciprocal space refinement methods, to name a few. Therefore, the running time of the procedure can range from seconds or minutes on a PC to weeks or months on a supercomputer. Trial structures are continually and simultaneously processed, with the final figure-of-merit values of all structures stored in a file. The user can review a dynamic histogram during the processing of the trials in order to determine whether or not a solution is likely present in the set of completed trial structures.

2. Genetic algorithms

Genetic Algorithms (GAs) were developed by Holland [15] and are based on natural selection and population genetics. Traditional optimization methods focus on developing a solution from a single trial, whereas genetic algorithms operate with a *population* of candidate solutions.

We propose to use a GA to determine an efficient set of *SnB* input parameters in an effort to reduce the time-to-solution for determining a molecular crystal structure from X-ray diffraction data. We use a population of candidate *SnB* input parameters. Each member of the population is represented as a string in the population and a fitness function is used to assign a fitness (quality) value for each member. The members in the population obtain their fitness values by executing the *SnB* program with the input parameter values represented by their strings. Using "survival-of-the-fittest" selection, strings from the *old* population are used to create a *new* population based on their fitness values. The member strings selected can recombine using crossover and/or mutation operators. A crossover operator creates a new member by exchanging substrings between two candidate members, whereas a mutation operator randomly modifies a piece of an existing candidate. This procedure of combining and randomly perturbing member strings has, in many cases, been shown to produce stronger (i.e., more fit) populations as a function of time (i.e., number of generations).

Sugal [16] (sequential execution) and PGAPack [17,18] (parallel and sequential execution) genetic algorithm libraries were used in our work. The Sugal library provided a sequential GA and has additional capabilities, including a restart function that proved to be very important when determining fitness values for large molecular structures. The PGAPack library provided a parallel master/slave MPICH/MPI implementation that proved very efficient on distributed- and shared-memory ACDC-Grid compute platforms. Other key features include C and Fortran interfaces, binary-, integer-, real-, and character-valued native data types, object-oriented design, and multiple choices for GA operators and parameters. In addition, PGAPack is quite extensible. The PGAPack library was extended to include restart functionality and is currently the only library used for the ACDC-Grid production work.

3. *SnB* input parameters

The *SnB* computer program has approximately 100 input parameters, though not all parameters can be optimized. For the purpose of this study, 17 parameters were identified for participation in the optimization procedure. The *SnB* parameter names and brief descriptions follow.

1. NUM_REF: number of reflections used for invariant generation and phase determination.
2. RESO_MAX: minimum data resolution.
3. E_SIG_CUT: E/Sigma(E) > Cut.
4. NUM_INV: number of three-phase invariants to generate and utilize during the *Shake-and-Bake* procedure.
5. NUM_CYCLE: number of *Shake-and-Bake* cycles performed on every trial structure.
6. PH_REFINE_METHOD: fast parameter shift, slow parameter shift, tangent formula method.
7. PS_INIT_SHIFT: parameter shift angle in degrees.
8. PS_NUM_SHIFT: maximum number of angular shift steps.
9. PS_NUM_ITER: maximum number of parameter shift passes through phase list.
10. TAN_NUM_ITER: maximum number of passes through phase list when PH_REFINE_METHOD is set to tangent formula method.

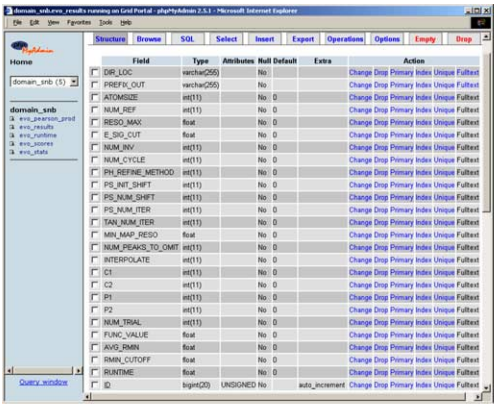
11. MIN_MAP_RESO: Fourier grid map resolution.
12. NUM_PEAKS_TO_OMIT: number of peaks to omit.
13. INTERPOLATE: a Boolean value that specifies whether or not to *interpolate* the density map.
14. C1: cycle 1 start.
15. C2: cycle 2 end.
16. P1: number of peaks to pick.
17. P2: number of heavy atom peaks to pick.

Eight known molecular structures were initially used to evaluate the genetic algorithm evolutionary molecular structure determination framework performance. These structures are 96016c [19], 96064c [20], crambin [21], Gramicidin A [22], Isoleucinomycin [23], pr435 [24], Triclinic Lysozyme [25], and Triclinic Vancomycin [26].

In order to efficiently utilize the computational resources of the ACDC-Grid, an accurate estimate must be made in terms of the resource requirements for *SnB* jobs that are necessary for the GA optimization. This includes runs with varying parameter sets over the complete set of eight known structures from our initial database.

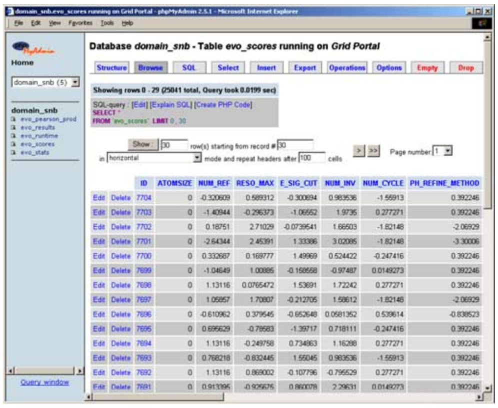
This was accomplished as follows. First, a small number of jobs were run in order to determine the required running time for each of the necessary jobs. Typically, this consisted of running a single trial for each of the jobs in order to predict the time required for the required number of trials for the job under consideration.

Approximately 25,000 population members were evaluated for the eight known molecular structures and stored in a MySQL [27] database table (evo_results). PhpMyAdmin [28] is a tool written in PHP for administration and use of MySQL over the web. The phpMyAdmin interface was used to display the evo_results database table, as shown in Fig. 3.


domain_snb.evo_results running on Grid Portal - phpMyAdmin 2.5.1 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Structure Browse SQL Select Insert Export Operations Options Empty Drop
Home
domain_snb (5)
domain_snb
evo_pearson_prod
evo_results
evo_runtime
evo_scores
evo_stats
Query window

	Field	Type	Attributes	Null	Default	Extra	Action
□	DIR_LOC	varchar(255)		No			Change Drop Primary Index Unique Fulltext
□	PREFIX_OUT	varchar(255)		No			Change Drop Primary Index Unique Fulltext
□	ATOMSIZE	int(11)		No	0		Change Drop Primary Index Unique Fulltext
□	NUM_REF	int(11)		No	0		Change Drop Primary Index Unique Fulltext
□	RESO_MAX	float		No	0		Change Drop Primary Index Unique Fulltext
□	E_SIG_CUT	float		No	0		Change Drop Primary Index Unique Fulltext
□	NUM_INV	int(11)		No	0		Change Drop Primary Index Unique Fulltext
□	NUM_CYCLE	int(11)		No	0		Change Drop Primary Index Unique Fulltext
□	PH_REFINE_METHOD	int(11)		No	0		Change Drop Primary Index Unique Fulltext
□	PS_INIT_SHIFT	int(11)		No	0		Change Drop Primary Index Unique Fulltext
□	PS_NUM_SHIFT	int(11)		No	0		Change Drop Primary Index Unique Fulltext
□	PS_NUM_ITER	int(11)		No	0		Change Drop Primary Index Unique Fulltext
□	TAN_NUM_ITER	int(11)		No	0		Change Drop Primary Index Unique Fulltext
□	MIN_MAP_RESO	float		No	0		Change Drop Primary Index Unique Fulltext
□	NUM_PEAKS_TO_OMIT	int(11)		No	0		Change Drop Primary Index Unique Fulltext
□	INTERPOLATE	int(11)		No	0		Change Drop Primary Index Unique Fulltext
□	C1	int(11)		No	0		Change Drop Primary Index Unique Fulltext
□	C2	int(11)		No	0		Change Drop Primary Index Unique Fulltext
□	P1	int(11)		No	0		Change Drop Primary Index Unique Fulltext
□	P2	int(11)		No	0		Change Drop Primary Index Unique Fulltext
□	NUM_TRIAL	int(11)		No	0		Change Drop Primary Index Unique Fulltext
□	FUNC_VALUE	float		No	0		Change Drop Primary Index Unique Fulltext
□	AVG_RMIN	float		No	0		Change Drop Primary Index Unique Fulltext
□	RMIN_CUTOFF	float		No	0		Change Drop Primary Index Unique Fulltext
□	RUNTIME	float		No	0		Change Drop Primary Index Unique Fulltext
□	ID	bigint(20)	UNSIGNED	No		auto_increment	Change Drop Primary Index Unique Fulltext

Fig. 3. MySQL database table for *SnB* trial results.



Database *domain_snb* - Table *evo_scores* running on *Grid Portal*

Showing rows 0 - 29 (25041 total, Query took 0.0199 sec)

SQL-query : [Edit] [Explain SQL] [Create PHP Code]

```
SELECT *
FROM `evo_scores` LIMIT 0 , 30
```

		ID	ATOMSIZE	NUM_REF	RESO_MAX	E_SIG_CUT	NUM_INV	NUM_CYCLE	PH_REFINE_METHOD
Edit	Delete	7704	0	-0.320609	0.589312	-0.300694	0.983536	-1.55913	0.392246
Edit	Delete	7703	0	-1.40944	-0.296373	-1.06552	1.9735	0.277271	0.392246
Edit	Delete	7702	0	0.18751	2.71029	-0.0739541	1.66503	-1.82148	-2.06929
Edit	Delete	7701	0	-2.64344	2.45391	1.33386	3.02085	-1.82148	-3.30006
Edit	Delete	7700	0	0.332687	0.169777	1.49969	0.524422	-0.247416	0.392246
Edit	Delete	7699	0	-1.04649	1.00885	-0.158558	-0.97487	0.0149273	0.392246
Edit	Delete	7698	0	1.13116	0.0765472	1.53691	1.72242	0.277271	0.392246
Edit	Delete	7697	0	1.05857	1.70807	-0.212705	1.58612	-1.82148	-2.06929
Edit	Delete	7696	0	-0.610962	0.379545	-0.652648	0.0581352	0.539614	-0.838523
Edit	Delete	7695	0	0.695629	-0.78583	-1.39717	0.718111	-0.247416	0.392246
Edit	Delete	7694	0	1.13116	-0.249758	0.734863	1.16288	0.277271	0.392246
Edit	Delete	7693	0	0.768218	-0.832445	1.55045	0.983536	-1.55913	0.392246
Edit	Delete	7692	0	1.13116	0.869002	-0.107796	-0.795529	0.277271	0.392246
Edit	Delete	7691	0	0.913395	-0.925675	0.860078	2.29631	0.0149273	0.392246

Fig. 4. Standard scores for Pearson product–moment correlation coefficient calculations.

From these trial results, the mean ($\overline{X}^j$) and standard deviations (s^j) were calculated for each input parameter j and used to determine the standard scores (z_i^j) for each trial i,

$$z_i^j = \frac{X_i^j - \overline{X^j}}{s^j},$$

for all i and j where the trial parameter value for trial i and parameter j is X_i^j. Fig. 4 shows the standard scores of the parameters under consideration.

The Pearson product–moment correlation coefficients (r_k^j) are calculated for input parameter j and molecular structure k by

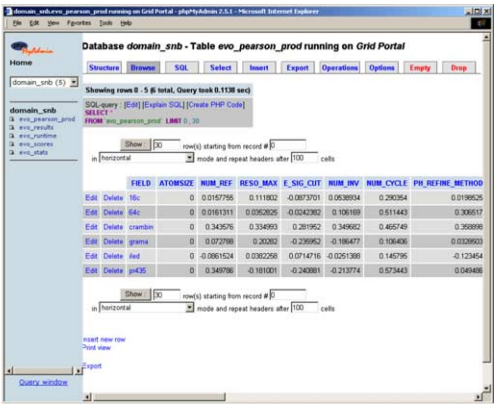
$$r_k^j = \frac{\sum z_k^j z_k^{\text{runtime}}}{N - 1},$$

for all j and k, where N denotes the degrees of freedom and z_k^{runtime} represents the standard score of the GA trial run time. Refer to Fig. 5.

The input parameters that have the largest absolute magnitude Pearson product–moment correlation coefficient with respect to the observed trial run times are selected and used to form a predictive run time function that is fit using a linear least squares routine

$$X_i^{\text{runtime}} = \sum a_j r_k^j \overline{X^j},$$

where the observed X_i^{runtime} trial run time is fit to a selected sub-set of input parameter values j, $\overline{X^j}$ denotes the input parameter value, r_k^j denotes the respective molecular structure k Pearson product–moment correlation coefficient, and a_j denotes the linear least square fit coefficients for each j input parameter. We use this function within the grid-enabled data-mining infrastructure to estimate the maximum number of *SnB* GA generations and the maximum size of the population that would run on a given computational resource within the specified time frame.


domain_snb.evo_pearson_prod running on Grid Portal - phpMyAdmin 2.5.1 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Home
domain_snb (5)
domain_snb
evo_pearson_prod
evo_results
evo_runtime
evo_scores
evo_stats
Structure Browse SQL Select Insert Export Operations Options Empty Drop
Show :
Insert new row
Print view
Export
Query window

Database *domain_snb* - Table *evo_pearson_prod* running on *Grid Portal*

Showing rows 0 - 5 (6 total, Query took 0.1138 sec)

SQL-query : [Edit] [Explain SQL] [Create PHP Code]

```
SELECT *
FROM `evo_pearson_prod` LIMIT 0 , 30
```

Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

		FIELD	ATOMSIZE	NUM_REF	RESO_MAX	E_SIG_CUT	NUM_INV	NUM_CYCLE	PH_REFINE_METHOD
Edit	Delete	16c	0	0.0157755	0.111802	-0.0873701	0.0538934	0.290354	0.0198525
Edit	Delete	64c	0	0.0161311	0.0352825	-0.0242382	0.106169	0.511443	0.306517
Edit	Delete	crambin	0	0.343576	0.334993	0.281952	0.349682	0.465749	0.358898
Edit	Delete	grama	0	0.072788	0.20282	-0.235952	-0.186477	0.106406	0.0328503
Edit	Delete	iled	0	-0.0861524	0.0382258	0.0714716	-0.0251388	0.145796	-0.123454
Edit	Delete	pr435	0	0.349786	-0.181001	-0.240881	-0.213774	0.573443	0.049486

Show : 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

Fig. 5. Pearson product–moment correlation coefficient database table.

The ACDC-Grid infrastructure automatically updates the correlation coefficients based on the availability of new trial data appearing in the *SnB* trial result table. Thus, run time estimates for any given structure continually evolve throughout the GA optimization process.

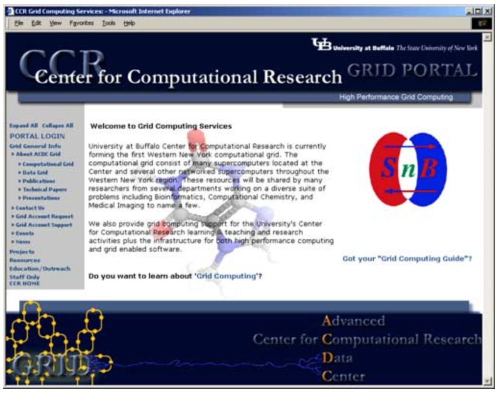
For example, if there are 50 processors available for 150 minutes on ACDC-Grid compute platform "A", we are interested in determining the maximum number of GA generations and the size of the population that can run on "A" and complete within 150 min. Based on this information, the data mining algorithms can make intelligent choices of not only which structures to evaluate, but they can completely define the *SnB* GA job that should be executed. This type of run time prediction is an essential component of our system for providing a level of quality of service. Further, in our experience, this type of run time parameter-based prediction is almost always necessary when queue managed computational resources are employed.

4. *Shake-and-Bake* grid-enabled data mining

The *SnB* grid-enabled data mining application utilizes the ACDC-Grid infrastructure and web portal, as shown in Fig. 6.

A typical *SnB* job uses the Grid Portal to supply the molecular structures parameter sets to optimize, the data file metadata, the grid-enabled *SnB* mode of operation (dedicated or back fill), and the *SnB* termination criteria. This information can be provided via the point and click web portal interface or by specifying a batch script, as shown in Fig. 7.

The database job script can accept command line arguments and can be activated or de-activated at any time by adjusting the database job grid portal parameters. A fully configurable time interval is used by the grid portal to execute some or all of the configured database jobs (normally this time interval is set to 10 min).



Welcome to Grid Computing Services

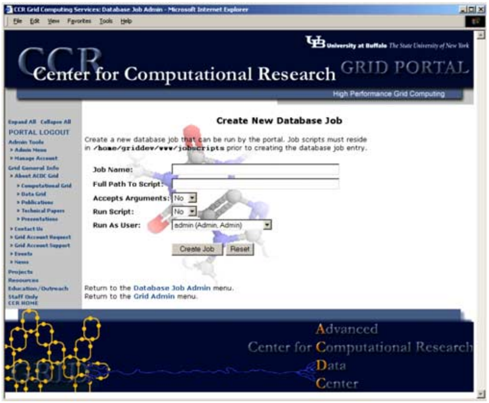
University at Buffalo Center for Computational Research is currently forming the first Western New York computational grid. The computational grid consist of many supercomputers located at the Center and several other networked supercomputers throughout the Western New York region. These resources will be shared by many researchers from several departments working on a diverse suite of problems including Bioinformatics, Computational Chemistry, and Medical Imaging to name a few.

We also provide grid computing support for the University's Center for Computational Research learning & teaching and research activities plus the infrastructure for both high performance computing and grid enabled software.

Got your "Grid Computing Guide"?

Do you want to learn about 'Grid Computing'?

Fig. 6. The ACDC-Grid web portal user interface.



Create New Database Job

Create a new database job that can be run by the portal. Job scripts must reside in /home/griddev/www/jobscripts prior to creating the database job entry.

Job Name:
Full Path To Script:
Accepts Arguments: No
Run Script: No
Run As User: admin (Admin, Admin)

Return to the Database Job Admin menu.
Return to the Grid Admin menu.

Fig. 7. The ACDC-Grid web portal database job interface.

The Grid Portal then assembles the required *SnB* application data and supporting files, execution scripts, database tables, and submits jobs for parameter optimization based on the current database statistics. ACDC-Grid job management automatically determines the appropriate execution times, number of trials, number of processors for each available resource, as well as logging and status of all concurrently executing resource jobs. In addition, it automatically incorporates the *SnB* trial results into the molecular structure database, and initiates post-processing of the updated database for subsequent job submissions. Fig. 8 shows the logical relationship for the *SnB* grid-enabled data mining routine described.

For example, starting September 8, 2003, a backfill data mining *SnB* job was activated at the Center for Computational Research using the ACDC-Grid computational and data grid resources. The ACDC-Grid historical job-monitoring infrastructure is used to obtain the jobs completed for the period of September 8, 2003 to January 10, 2004, as shown in Fig. 9.

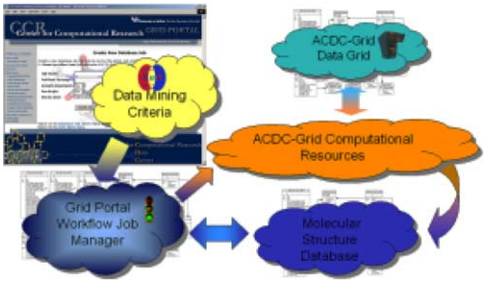
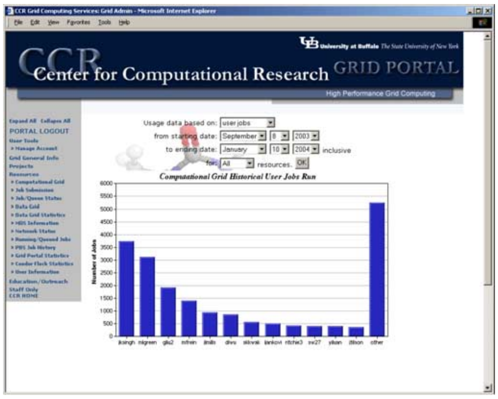
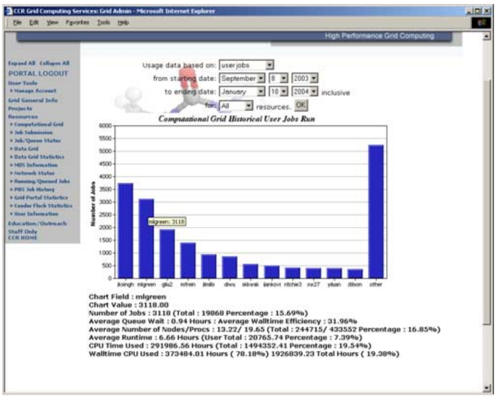

ACDC-Grid Data Grid
Data Mining Criteria
ACDC-Grid Computational Resources
Grid Portal Workflow Job Manager
Molecular Structure Database

Fig. 8. ACDC-Grid grid-enabled data mining diagram.



Usage data based on: user jobs
from starting date: September 8 2003
to ending date: January 10 2004 inclusive
for: All resources.

Fig. 9. ACDC-Grid job monitoring information for all resources and users.



Usage data based on: user jobs
from starting date: September 8 2003
to ending date: January 10 2004 inclusive
for: All resources.

```
Chart Field : mlgreen
Chart Value : 3118.00
Number of Jobs : 3118 (Total : 19868 Percentage : 15.69%)
Average Queue Wait : 0.94 Hours : Average Walltime Efficiency : 31.96%
Average Number of Nodes/Procs : 13.22/ 19.65 (Total : 244715/ 433552 Percentage : 16.85%)
Average Runtime : 6.66 Hours (User Total : 20765.74 Percentage : 7.39%)
CPU Time Used : 291986.56 Hours (Total : 1494352.41 Percentage : 19.54%)
Walltime CPU Used : 373484.01 Hours ( 78.18%) 1926839.23 Total Hours ( 19.38%)
```

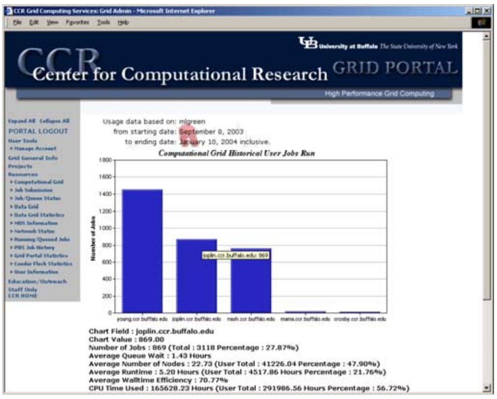
Fig. 10. ACDC-Grid job monitoring statistics for user mlgreen.

The activated data mining *SnB* job template is being run by user mlgreen. By hovering over the bar in the chart, as shown in Fig. 10, one can see mlgreen's job statistics. Further, notice that 3118 jobs have been completed on the ACDC-Grid resources over this time period. The ACDC-Grid job monitoring also dynamically reports job statistics for the data mining jobs. The total number of jobs completed by all users on all resource is 19,868 where the data mining jobs represent 15.69% of the total. The average number of processes for a data-mining job was 19.65 and the total number of processors used over this period was 433,552, where the data mining jobs accounted for 16.85% of the total. The data mining jobs consumed 291,987 CPU hours, which was 19.54% of the total CPU hours consumed (1,494,352 CPU hours).

A subsequent mouse click on the bar chart provides additional information characterizing in more detail the jobs completed by user mlgreen. Here, we see five computational resources that processed the 3118 data mining jobs. The statistics for the Joplin compute platform are shown in Fig. 11. Note that all statistics are based only on the jobs completed by the mlgreen user. There were 869 jobs processed by the Joplin compute platform representing 27.87% of the 3118 data mining jobs.

Clicking on the bar chart drills down into a full description of all jobs processed by the Joplin compute platform, as shown in Fig. 12. The information presented includes job ID, username, group name, queue name, node count, processes per node, queue wait time, wall time used, wall time requested, wall time efficiency, CPU time, physical memory used, virtual memory used, and job completion time/date.

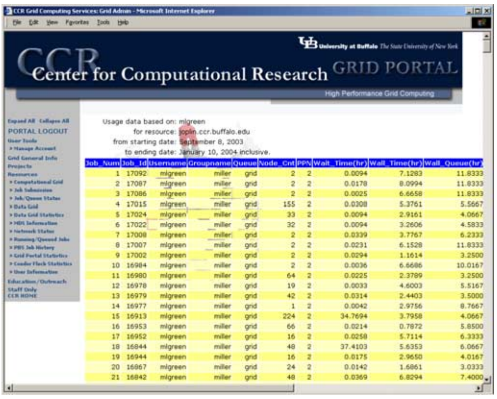
The ACDC-Grid data mining backfill mode of operation only uses computational resources that are currently not scheduled for use by the native queue scheduler. These resources are commonly referred to as "backfill jobs," as users can run jobs on the associated nodes without affecting the queued jobs. Many queues and schedulers give this information in terms of the number of nodes available and the time for which the nodes are available. The ACDC-Grid infrastructure monitors this infor-



Usage data based on: mlgreen
from starting date: September 8, 2003
to ending date: January 10, 2004 inclusive.

Chart Field : joplin.ccr.buffalo.edu
Chart Value : 869.00
Number of Jobs : 869 (Total : 3118 Percentage : 27.87%)
Average Queue Wait : 1.43 Hours
Average Number of Nodes : 22.73 (User Total : 41226.04 Percentage : 47.90%)
Average Runtime : 5.20 Hours (User Total : 4517.86 Hours Percentage : 21.76%)
Average Walltime Efficiency : 70.77%
CPU Time Used : 165628.23 Hours (User Total : 291986.56 Hours Percentage : 56.72%)

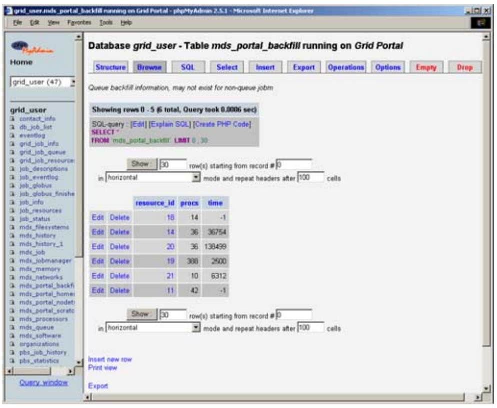
Fig. 11. ACDC-Grid job monitoring statistics for user mlgreen.



Usage data based on: mlgreen
for resource: joplin.ccr.buffalo.edu
from starting date: September 8, 2003
to ending date: January 10, 2004 inclusive.

Job_Num	Job_Id	Username	Groupname	Queue	Node_Cnt	PPN	Wait_Time(hr)	Wall_Time(hr)	Wall_Queue(hr)
1	17092	mlgreen	miller	grid	2	2	0.0094	7.1283	11.8333
2	17087	mlgreen	miller	grid	2	2	0.0178	8.0994	11.8333
3	17086	mlgreen	miller	grid	2	2	0.0025	6.6658	11.8333
4	17015	mlgreen	miller	grid	155	2	0.0308	5.3761	5.5667
5	17024	mlgreen	miller	grid	33	2	0.0094	2.9161	4.0667
6	17022	mlgreen	miller	grid	32	2	0.0094	3.2606	4.5833
7	17008	mlgreen	miller	grid	2	2	0.0339	3.7767	6.2333
8	17007	mlgreen	miller	grid	2	2	0.0231	6.1528	11.8333
9	17002	mlgreen	miller	grid	2	2	0.0294	1.1614	3.2500
10	16984	mlgreen	miller	grid	2	2	0.0036	6.6686	10.0167
11	16980	mlgreen	miller	grid	64	2	0.0225	2.3789	3.2500
12	16978	mlgreen	miller	grid	19	2	0.0033	4.6003	5.5167
13	16979	mlgreen	miller	grid	42	2	0.0314	2.4403	3.5000
14	16977	mlgreen	miller	grid	1	2	0.0042	2.9756	8.7667
15	16913	mlgreen	miller	grid	224	2	34.7694	3.7958	4.0667
16	16953	mlgreen	miller	grid	66	2	0.0214	0.7872	5.8500
17	16952	mlgreen	miller	grid	16	2	0.0258	5.7114	6.3333
18	16844	mlgreen	miller	grid	48	2	37.4103	5.6353	6.0667
19	16944	mlgreen	miller	grid	16	2	0.0175	2.9650	4.0167
20	16867	mlgreen	miller	grid	24	2	0.0142	1.6861	3.0333
21	16842	mlgreen	miller	grid	48	2	0.0369	6.8294	7.4000

Fig. 12. ACDC-Grid job monitoring tabular accounting of completed job statistics.



Database *grid_user* - Table *mds_portal_backfill* running on *Grid Portal*

Queue backfill information, may not exist for non-queue jobm

Showing rows 0 - 5 (6 total, Query took 0.0006 sec)

```
SQL-query : [Edit] [Explain SQL] [Create PHP Code]
SELECT *
FROM 'mds_portal_backfill' LIMIT 0 , 30
```

Show 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

		resource_id	procs	time
Edit	Delete	18	14	-1
Edit	Delete	14	36	36754
Edit	Delete	20	36	138499
Edit	Delete	19	388	2500
Edit	Delete	21	10	6312
Edit	Delete	11	42	-1

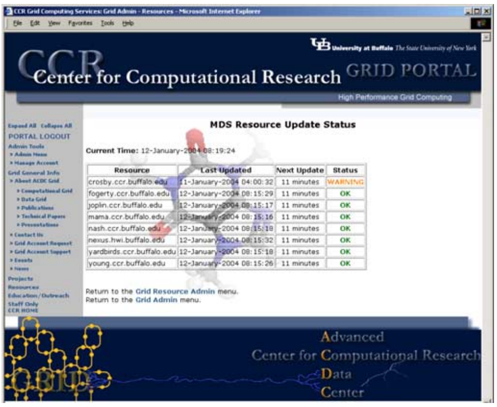
Show 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

Fig. 13. ACDC-Grid backfill information for all resources.

mation for all of the computational resources and stores this information in a MySQL database table, as shown in Fig. 13.

Fig. 13 also shows the number of processors and wall time that are available for each resource. Note that a value of −1 for the available wall time represents an unlimited amount of time (no currently queued job require the use of these processors). The activated data mining template can obtain the number of processors and wall time available for a given compute platform and then check the status of the platform before determining the actual GA *SnB* data mining job parameters (see Figs. 14 and 15).

Using the Pearson product–moment fit function derived earlier, the new data mining job run time is estimated based on the current ACDC-Grid *SnB* molecular structure database information.



MDS Resource Update Status

Current Time: 12-January-2004 08:19:24

Resource	Last Updated	Next Update	Status
crosby.ccr.buffalo.edu	11-January-2004 04:00:32	11 minutes	WARNING
fogerty.ccr.buffalo.edu	12-January-2004 08:15:29	11 minutes	OK
joplin.ccr.buffalo.edu	12-January-2004 08:15:17	11 minutes	OK
mama.ccr.buffalo.edu	12-January-2004 08:15:16	11 minutes	OK
nash.ccr.buffalo.edu	12-January-2004 08:15:18	11 minutes	OK
nexus.hwi.buffalo.edu	12-January-2004 08:15:32	11 minutes	OK
yardbirds.ccr.buffalo.edu	12-January-2004 08:15:18	11 minutes	OK
young.ccr.buffalo.edu	12-January-2004 08:15:26	11 minutes	OK

Return to the Grid Resource Admin menu.
Return to the Grid Admin menu.

Fig. 14. ACDC-Grid computational resource status monitor.

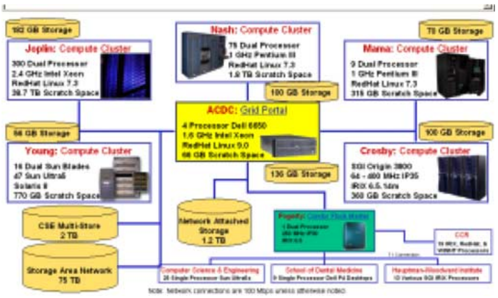


Fig. 15. ACDC-Grid data grid and computational grid integration.

Once the computational aspects of the data-mining job have been defined, the location of the required data files is determined by querying the ACDC-Grid Data Grid database. The ACDC-Grid Data Grid denotes a large network of distributed storage resources such as archival systems, Storage Area Networks, Network Attached Storage, and databases, which are linked logically creating global and persistent disk storage that can be accessed by all computational resources (refer to Fig. 15).

The data grid is designed to provide transparent management of data distributed across heterogeneous resources, such that the data is accessible via a uniform web interface and directly through a well-defined API. In addition, the data grid infrastructure enables the transparent migration of data between various resources while maintaining metadata information about each file and its location via a global database table. The system periodically migrates files between machines based on user patterns in order to achieve an efficient usage of resources.

The implementations of basic file management functions accessible via a platform-independent web interface provide the following features.

- User-friendly menus/interface.
- File Upload/Download to and from the Data Grid Portal.
- Simple web-based file editor.
- Efficient search utility.
- Logical display of files for a given user in three divisions (user/ group/ public).
  - Hierarchical.
  - List-based.
  - Three divisions: (user/ group/ public).
  - Sorting capability based on file metadata, i.e. filename, size, modification time, etc.
- Support multiple accesses to files in the data grid (file locking and synchronization primitives for version control).

Integrated security procedures allow authentication and authorization of users for data grid file access and enforce policies for data access and publishing. The gathering and display of statistical information on the data grid usage is automatically obtained through the data grid infrastructure. This information is particularly useful to administrators for optimizing the usage of resources. The data mining job template is then executed, leading to the migration and submission of the designed data-mining job and data files to the respective ACDC-Grid computational resource.

The activated data-mining template has two options of stopping criteria.

1. Continue submitting *SnB* data-mining application jobs until the optimal parameters have been found based on predetermined criteria.
2. Continue indefinitely (the data mining template is manually de-activated by the user when optimal parameters are found).

This illustrative example summarizes the evolutionary molecular structure determination optimization of the *Shake-and-Bake* method as instantiated in the *SnB* computer program.

5. Further remarks

The evolutionary strategy presented in this paper was used to substantially improve the performance of the *SnB* procedure. The GA procedure that we described improved the time-to-solution of *SnB* by up to a factor of two on a number of representative data sets.

Many scientific applications can take advantage of an evolutionary procedure in an effort to optimize its input parameters. We have identified several such applications and are currently generalizing the grid-enabled data mining procedure as a grid-enabling application template.

We are currently working on several related projects, including enhancing the ACDC-Grid infrastructure to include predictive scheduling, intelligent data migration over several large data repositories, and lightweight hierarchical grid monitoring tools.

We plan to package and distribute much of this work in the near future.

Acknowledgments

Support was provided by the Center for Computational Research, the National Science Foundation under award ACI-02-04918, and the National Institutes of Health under grant GM-46733. The authors would like to thank Cynthia Cornelius, Steve Gallo, Martins Innus, Adam Koniak, and Jason Rappleye for their contributions. Additional information can be found at www.ccr.buffalo.edu/.

References

[1] M.L. Green, R. Miller, A Client-server prototype for application grid-enabling template design, Parallel Processing Letters 14 (2) (2004) 241–253.
[2] <http://www.acdc.ccr.buffalo.edu/>.
[3] http://www.cse.buffalo.edu/pub/www/faculty/miller/talks_html/sc2003/sc2003-panel.pdf.
[4] I. Foster, C. Kesselman, The Grid: Blueprint For a New Computing Infrastructure, Morgan Kauffman Publishers, Inc., San Francisco, 1999.
[5] F. Berman, G. Fox, T. Hey, Grid Computing: Making the Global Infrastructure a Reality, John Wiley & Sons, New York, 2003.
[6] R. Miller, S.M. Gallo, H.G. Khalak, C.M. Weeks, SnB: crystal structure determination via Shake-and-Bake, J. Appl. Crystallogr. 27 (1994) 613–621.
[7] C.M. Weeks, G.T. DeTitta, H.A. Hauptman, P. Thuman, R. Miller, Structure solution by minimal function phase refinement and Fourier filtering: II. implementation and applications, Acta Crystallogr. A 50 (1994) 210–220.
[8] G.T. DeTitta, C.M. Weeks, P. Thuman, R. Miller, H.A. Hauptman, Structure solution by minimal function phase refinement and Fourier filtering: theoretical basis, Acta Crystallogr. A 50 (1994) 203–210.
[9] H.A. Hauptman, H. Xu, C.M. Weeks, R. Miller, Exponential Shake-and-Bake: theoretical basis and applications, Acta Crystallogr. A 55 (1999) 891–900.
[10] C.M. Weeks, R. Miller, The design and implementation of SnB v2.0, J. Appl. Crystallogr. 32 (1999) 120–124.
[11] C.M. Weeks, R. Miller, Optimizing Shake-and-Bake for proteins, Acta Crystallogr. D 55 (1999) 492–500.
[12] J. Rappleye, M. Innus, C.M. Weeks, R. Miller, SnB v2.2: an example of crystallographic multiprocessing, J. Appl. Crystallogr. 35 (2002) 374–376.
[13] M.L. Green, R. Miller, Grid computing in Buffalo, Annals of the European Academy of Sciences, New York, 2003, pp. 191–218.
[14] H.A. Hauptman, A minimal principle in the phase problem, in: D. Moras, A.D. Podjarny, J.C. Thierry (Eds.), Crystallographic Computing: From Chemistry to Biology, vol. 5, IUCr Oxford University Press, Oxford, 1991, 324–332.
[15] J. Holland, Adaption in Natural and Artifical Systems, MIT Press, Cambridge, 1992.
[16] A. Hunter, SUGAL User Manual v2.0, available from: <http://www.dur.ac.uk/andrew1.hunter/Sugal/>.
[17] D. Levine. PGAPack, 1995. A public-domain parallel genetic algorithm library, available anonymous ftp from ftp.mcs.anl.gov in the directory pub/pgapack, file pgapack.tar.Z.

[18] D. Levine. Users guide to the PGAPack parallel genetic algorithm library. Technical Report ANL-95/18, Argonne National Laboratory, Mathematics and Computer Science Division, June 23, 1995.

[19] D. Ho, personal communication: C109 H73 N1.

[20] D. Ho, personal communication: C220 H148.

[21] M.G. Usha, R.J. Wittebort, Orientational ordering and dynamics of the hydrate and exchangeable hydrogen atoms in crystalline crambin, J. Mol. Biol. 208 (4) (1989) 669–678.

[22] D.A. Langs, G.D. Simth, C. Courseille, G. Precigoux, M. Hospital, Monoclinic uncomplexed double-stranded antiparallel left-handed ß$^{5.6}$-helix (↑↓ß$^{5.6}$) structure of gramicidin a: alternative patterns of helical association and deformation, Proc. Natl. Acad. Sci., USA 88 (1991) 5345–5349.

[23] V. Pletnev, N. Galitskii, G.D. Smith, C.M. Weeks, W.L. Duax, Crystal and molecular structure of Isoleucinomycin, cyclo[-(D-Ile–Lac–Ile–D-Hyi)3-] (C60H102N6O18), Biopolymers 19 (1980) 1517–1534.

[24] C.M. Weeks, W.L. Duax, 9α-Chlorocortison, an active cortisol derivative, Acta Cryst. B 30 (1974) 2516–2519.

[25] J.M. Hodsdon, G.M. Brown, L.C. Sieker, L.H. Jensen, Refinement of triclinic Lysozyme: I. Fourier and least-squares methods, Acta Crystallogr. B 46 (1990) 54–62.

[26] P.J. Loll, R. Miller, C.M. Weeks, P.H. Axelsen, A ligand-mediated dimerization mode for vancomycin, Chemistry and Biology 5 (1998) 293–298.

[27] MySQL Database Open Source Database, available from: <http://www.mysql.com/>.

[28] The phpMyAdmin Project, available from: <http://www.phpmyadmin.net/home_page/>.