Title

Advances in Rosetta protein structure prediction on massively parallel systems

Permalink
https://escholarship.org/uc/item/87g6q6bw

Journal
IBM Journal of Research and Development, 52(1)

ISSN
0018-8646

Authors
Raman, S.
Baker, D.
Qian, B.
et al.

Publication Date
2008

Peer reviewed

Advances in Rosetta protein structure prediction on massively parallel systems

S. Raman
B. Qian
D. Baker
R. C. Walker

One of the key challenges in computational biology is prediction of three-dimensional protein structures from amino-acid sequences. For most proteins, the "native state" lies at the bottom of a free-energy landscape. Protein structure prediction involves varying the degrees of freedom of the protein in a constrained manner until it approaches its native state. In the Rosetta protein structure prediction protocols, a large number of independent folding trajectories are simulated, and several lowest-energy results are likely to be close to the native state. The availability of hundred-teraflop, and shortly, petaflop, computing resources is revolutionizing the approaches available for protein structure prediction. Here, we discuss issues involved in utilizing such machines efficiently with the Rosetta code, including an overview of recent results of the Critical Assessment of Techniques for Protein Structure Prediction 7 (CASP7) in which the computationally demanding structure-refinement process was run on 16 racks of the IBM Blue Gene/L™ system at the IBM T. J. Watson Research Center. We highlight recent advances in high-performance computing and discuss future development paths that make use of the next-generation petascale (>10^{12} floating-point operations per second) machines.

Introduction

With genetic sequencing completed for the entire genome of a number of organisms including humans, the next challenge is to functionally characterize the proteins encoded by the genes and to understand their roles and interactions in cellular pathways. High-resolution three-dimensional (3D) protein structures can help in understanding biological functions of proteins and also explain the underlying molecular mechanisms of protein interactions. The availability of high-resolution protein structures should significantly accelerate efforts toward targeted drug development.

It has been known for more than 40 years that the 3D structures of proteins under normal physiological conditions are uniquely determined by the composition of their amino acids, which form a sequence called the *primary structure* of a protein [1]. However, despite considerable technical advances, the experimental determination of protein structures by nuclear magnetic resonance (NMR) and x-ray diffraction techniques remains slow, expensive, and arduous. In particular, the rate at which protein structures are being experimentally solved is lagging far behind the explosive rate at which protein sequence information is being gathered by high-throughput genome sequencing efforts. Thus, given an amino-acid sequence, a high-throughput methodology to computationally predict protein structures at atomic-level accuracy is one of the long-standing challenges in computational biology.

If the protein of interest shares a reasonable sequence similarity with a protein of known structure, then the latter can be used as a template for modeling. If, however, no detectable templates exist or the similarity is small, then *de novo* techniques have to be used [2]. In *de novo* methods, a protein structure is predicted in the absence of any tertiary (i.e., 3D) structure information. In other words, no known proteins can be used as templates for the starting model. The ultimate objective is to make high-resolution protein structures readily available for all proteins of biological interest and then to extend this to enable the design of man-made proteins.

0018-8646/08/$5.00 © 2008 IBM

Predicting protein structures presents a formidable problem given the large dimensionality of the protein conformation search space [3]. For instance, in the Rosetta approach described in the next section, about 1,000 degrees of freedom exist for a 150-amino-acid protein. The degrees of freedom consist of backbone and side-chain torsion angles. In addition, the problem is further complicated by the presence of a large number of local minima in the energy landscape because of the atomic repulsions at close distances. Therefore, the task is computationally very demanding and requires a significant amount of computing power to sample a sufficiently large ensemble of conformations in the protein conformation space to locate the *native state*, that is, the operative or functional form of the protein.

The Rosetta method

Protein structure prediction and refinement are carried out using the open-source software Rosetta, developed in David Baker's research group at the University of Washington. Rosetta is a software package and a protein structure prediction algorithm. In this paper, when referring to *Rosetta*, we refer to the algorithm as implemented within the Rosetta software package.

At the core of the Rosetta algorithm are potential energy functions for computing the interactions within and between macromolecules, as well as optimization methods for determining the lowest-energy structure for an amino-acid sequence. While significant insight can be gained about protein topology from *de novo* models and medium-resolution atomic interactions from homology models, the atomic details that are responsible for protein function and interaction require models at the highest resolution. This motivated us to develop the high-resolution refinement protocol in Rosetta. The starting points for this protocol are approximate models of the desired structure. These models come as two forms. First, they may be homology models, generated by comparing the amino-acid sequence of the protein whose structure is to be predicted with similar sequences of known structures and then accounting for the differences in the sequence with predictions based on patterns detected in the available experimental data. Second, the models may come from low-resolution NMR refinement of the protein under study or similar proteins. Typically, only a handful of initial starting structures are available, and these may be significantly perturbed from the true native structure. Thus, we adopt a two-step approach consisting of low-resolution protein-backbone loop modeling followed by high-resolution refinement. This approach works in a step-wise pattern whereby the initial structure is perturbed using a set of rules that operate on a low-resolution representation of the structure. This has the effect of generating new low-resolution model structures. Then, each of these structures is converted to a high-resolution structure (containing all atoms), and the energy is evaluated using the Rosetta energy function as described later. This structure then undergoes a series of optimization steps (high-resolution optimization), as described below, that attempt to minimize the energy as described by the Rosetta energy function. Assuming sufficient conformation space has been sampled, the structure with the lowest energy is likely to be close to the native structure. The individual steps are described in more detail below.

Low-resolution loop modeling—In protein structure prediction, it is critical to rapidly sample configuration space diversely enough to have sufficient probability of generating a model that is close to the native structure. This is achieved in the low-resolution loop-modeling phase. In this step, the protein has a reduced representation known as the *low-resolution mode*. At this stage, we introduce structural changes to selected regions of the protein structure by rebuilding them from a customized set of fragment libraries [4]. The fixed regions in the initial structure impose a geometrical constraint on the newly built regions, ensuring fidelity of the new structures to the initial overall folding, without large backbone conformation differences. This step introduces a large diversity in structural conformation while preserving the folding.

High-resolution optimization—All the atoms of the protein are explicitly represented, and the protein structure has all available backbone and side-chain degrees of freedom. The structural changes consist of perturbing the backbone conformation of stochastically selected regions and rapid combinatorial optimization of the side chains [5, 6] to accommodate the new backbone conformation, followed by gradient-based minimization to the nearest local energy minimum [7, 8].

After every perturbation in the low-resolution mode, or perturbation optimization in the high-resolution mode, the new conformation of the structure is accepted or rejected with the Boltzmann probability of the Metropolis Monte Carlo criterion [9]. A single trajectory of protein structure refinement involves hundreds of iterations of structure perturbation moves in the low- and high-resolution modes. At the end of the refinement trajectory, the model settles at a low minimum of the energy landscape. For a given protein, tens of thousands to hundreds of thousands of independent trajectories are simulated. A small fraction of these end up as local minima that are close in energy to the native structure [7].

The Rosetta high-resolution energy function

One of the critical aspects of high-resolution structure refinement is an energy function that can discriminate a native structure from a non-native structure at atomic

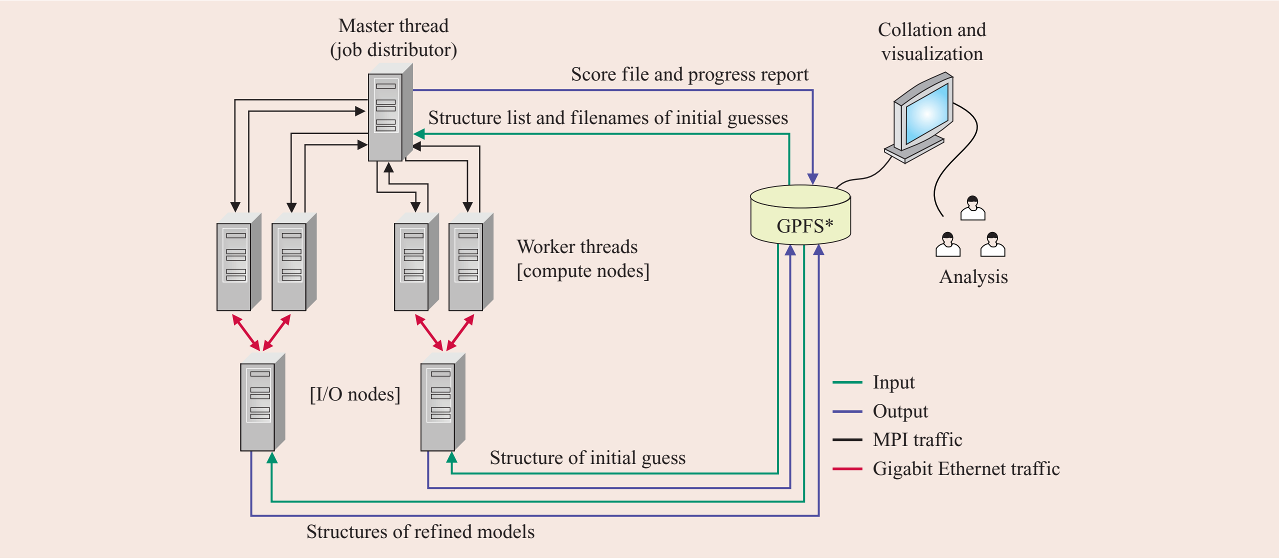
Master thread
(job distributor)
Score file and progress report
Structure list and filenames of initial guesses
Collation and visualization
GPFS*
Worker threads
[compute nodes]
Analysis
[I/O nodes]
Input
Output
MPI traffic
Gigabit Ethernet traffic
Structure of initial guess
Structures of refined models

Figure 1

Diagram showing communication patterns within the Rosetta code when carrying out a structure prediction on the IBM BG/L platform. A job distributor, running on the master node, controls load balancing and produces summary information about the prediction progress. The worker threads receive their computation instructions from the job distributor in batch form over a message-passing interface. See text for additional details. (GPFS: General Parallel File System*.)

resolution such that the native structure lies at the bottom of a free-energy landscape. The Rosetta energy function makes use of the 12-6 Lennard–Jones energy function [6], the Lazaridis–Karplus implicit solvation model [10], orientation-dependent hydrogen bonds derived from quantum mechanical calculations [11], and statistics computed on high-resolution x-ray structures in the Brookhaven Protein Data Bank (PDB) [12], a pair interaction term that represents longer-range electrostatic interactions between polar atoms and π–π and cation–π interactions, a side-chain torsional potential derived from the Dunbrack backbone-dependent rotamer library [13], and a backbone torsional potential dependent on secondary structure [14]. Each of the above terms can be pair-wise factorized, and the total energy of the structure is the weighted sum of all these terms. The functional forms of the components of the energy function are shown in **Table 1** [15]. In this table, *P* refers to the probability of specified input conditions, such as backbone torsion angles, occurring in databases of experimental results. These probability tables have been constructed by fitting to high-resolution structures that have been determined experimentally and are published in the PDB. The term *residue* in the table refers to an amino acid in the protein.

Running Rosetta on supercomputers

High-resolution structure refinement is a very computationally intensive task. In essence, the problem is one of global optimization of the free energy of the protein by varying all degrees of freedom in order to explore a highly complex multidimensional energy landscape. Each simulation comprises thousands of Monte Carlo minimization trajectories, each initiated from a distinct random number seed. Given that the high-resolution prediction of a protein structure of fewer than 200 amino acids could take thousands of CPU days, this problem (coupled with the need to minimize the wall-clock [elapsed] time for Critical Assessment of Techniques for Protein Structure Prediction 7 [CASP7] and related protein structure prediction benchmarks [16, 17]) is thus ideally suited for high-performance computing (HPC) on massively parallel systems.

Utilizing the Blue Gene/L system for structure prediction

The coarse-grain nature of the Rosetta structure refinement calculations makes the IBM Blue Gene/L* (BG/L) platform, with its large number of relatively low-power processors, the ideal architecture for protein structure prediction. We parallelized the Rosetta software by using a master/worker thread approach, using the message-passing interface (MPI) [18]. A schematic of the implementation is shown in **Figure 1**. In this approach, the master thread runs as a job distribution node that distributes packets of work to each of the worker threads. These threads then independently work on this packet and

Table 1 The components of the Rosetta high-resolution energy function.

Function name	*Functional form*	*Parameters*	*Description*
Ramachandran torsion preferences	$\sum_i -\ln[P(\phi_i, \psi_i \mid aa_i, ss_i)]$	i = residue index ϕ, ψ = backbone torsion angles aa = amino-acid type ss = secondary structure type	Based on the secondary structure of proteins, the backbone torsion angles assume preferred values. This term penalizes deviations from the preferred state represented by the Ramachandran map.
Lennard–Jones interactions	$\sum_i \sum_{j>i} \begin{cases} \left[\left(\frac{r_{ij}}{d_{ij}}\right)^{12} - 2\left(\frac{r_{ij}}{d_{ij}}\right)^{6}\right] e_{ij}, & \text{if } \frac{d_{ij}}{r_{ij}} > 0.6 \\ \left[-8759.2\left(\frac{d_{ij}}{r_{ij}}\right) + 5672.0\right] e_{ij}, & \text{else} \end{cases}$	i, j = residue indices d = interatomic distance e = geometric mean of atom well depths r = summed van der Waals radii	Short-range atomic interactions are dominated by attractive van der Waals forces. If the atoms get too close, they clash and are repelled. This term captures the attractive and repulsive interactions.
Hydrogen bonding	$\sum_i \sum_j (-\ln[P(d_{ij} \mid h_j ss_{ij})] - \ln[P(\cos\theta_{ij} \mid d_{ij} h_j ss_{ij})] - \ln[P(\cos\psi_{ij} \mid d_{ij} h_j ss_{ij})])$	i = donor residue index j = acceptor residue index d = acceptor–proton interatomic distance h = hybridization (sp2, sp3) ss = secondary structure type θ = proton–acceptor–acceptor base bond angle ψ = donor–proton–acceptor bond angle	The hydrogen bonds involve two groups: a donor and an acceptor group. These bonds are formed at very specific angles and distances between these groups. This is a statistical potential that scores a given conformation in comparison to hydrogen bonds observed in high-resolution crystal structures.
Solvation	$\sum_i \left[\Delta G_i^{\text{ref}} - \sum_j \left(\frac{2\,\Delta\, G_i^{\text{free}}}{4\,\pi^{3/2}\lambda_i r_{ij}^2} e^{-d_{ij}^2} V_j + \frac{2\,\Delta G_i^{\text{free}}}{4\,\pi^{3/2}\lambda_j r_{ij}^2} e^{-d_{ij}^2} V_i\right)\right]$	i, j = atom indices d = distance between atoms r = summed van der Waals radii λ = correlation length V= atomic volume ΔG^{ref}, ΔG^{free} = energy of a fully solvated atom	Naturally occurring proteins interact with the surrounding solvent. The interaction of the solvent with protein is called the *solvation-free energy* as described by this term.
Rotamer energy	$\sum_i -\ln\left[\frac{P(rot_i \mid \phi_i, \psi_i) P(aa_i \mid \phi_i, \psi_i)}{P(aa_i)}\right]$	i, j = residue indices rot = Dunbrack backbone-dependent rotamer aa = amino-acid type ϕ, ψ = backbone torsion angles	The side-chains of proteins adopt preferred conformations depending on the backbone state. This is a statistical potential derived from high-resolution crystal structures for scoring side-chain conformations.
Unfolded state reference energy	$\sum_{i=1}^{n} E_{\text{ref}}[aa_i]$	aa = amino-acid type n = number of residues E_{ref} = unfolded state reference energy for amino-acid type aa	Reference energy for every amino-acid type assuming unfolded state of the protein.

return to the master for more work. This approach has advantages because it deals with computational load-balancing issues due to different amounts of time required to refine each structure. The approach also has a disadvantage: With low numbers of processors, the overhead introduced by having a master thread that does not do any computation can have an impact on the overall efficiency. However, we have found that on most small computer clusters, because of the control of load balancing introduced by the job distributor model, the efficiency is only marginally affected by running one more MPI thread than there are available processors. On the BG/L platform, this is not possible because of the lack of multitasking support in the node OS (operating system), but here, jobs are typically run using 2,048 or more processors, in which case the efficiency impact of having one node run as a non-compute node is trivial. In addition, as the number of worker nodes increases (so the load on the master node increases), a point will be reached in which there is little to be gained by also running compute jobs on the master node.

Note that in Figure 1, the master thread uses MPI to send the filename for an initial structure guess (from a homology model or a low-resolution NMR experiment), as well as the number of trials to generate from this initial structure to each worker thread. Each worker thread then completes this batch of work, and each writes the results of its structure refinements in parallel to disk and communicates back to the master, using MPI, the summary information regarding the batch of refinements and a request for more work.

Overcoming problems with large processor counts

Early in our study, we encountered a number of challenges when running Rosetta on large numbers of processors (e.g., >8,192). First, the overall load on the master thread increases as a function of the number of processors; thus, at more than a certain number of processors, the master thread cannot handle the requests sufficiently fast, and the performance decreases. This was easily overcome by dynamically increasing the size of the work chunk (i.e., portion) that is given to each compute thread. However, in the future, it may become necessary to implement a hierarchical job distribution system with several job distributors that communicate with a single master job distribution thread in order to avoid such bottlenecks. This will be particularly important when utilizing processor counts in excess of 100,000 since this is the point at which the current load-balancing scheme will theoretically break down. This breakdown occurs because the number of individual refinement trajectories that are typically run for a given protein is on the order of 100,000 to 500,000. Thus, as the number of processors approaches the number of refinements to be run, the flexibility available for dynamic load balancing is reduced, as is the ability to regulate the load on the master node by modifying the work chunk size.

Even with code parallelization, the bottleneck in protein structure prediction is still the amount of conformational sampling that can be undertaken; thus, benefits exist for simply increasing the number of trajectories that are run. For the reasons discussed above, increasing the number of trajectories will address the issue of load balancing, at least in the short term, by increasing the amount of work that each processor is responsible for; however, long-term advancements, including more accurate predictions and the ability to treat larger proteins, will likely come from improving the complexity of the energy functional used and developing more efficient methods for sampling conformational space. In this regime, as compute power increases, the number of trajectories that can be run in a given time will remain roughly constant, and thus, it will be necessary to implement even finer-grain parallelism in order to ensure effective load balancing.

Another problem that was encountered early in this study concerned disk I/O issues. Initially, the job distributor thread was responsible for reading information from disk, packaging all the data required by the worker thread for computation and then sending this to the worker thread over MPI. The worker thread would then undertake the computation and routinely send progress reports back to the master thread so that it could write the status and necessary output files. Such an approach works fine on small clusters, but when moving to massively parallel systems, and in particular the BG/L system with its limited memory per node, this approach leads to severe performance degradation due to I/O and MPI bottlenecks at the master node. The I/O requirements of the Rosetta code are not particularly large, but once the computation moves to a large number of processors, the I/O rapidly becomes prohibitive. For example, a typical run for a protein of approximately 100 residues would have about 100 input structures totaling 100 KB each, as well as a database of parameters of approximately 65 MB. The completion of each of the trajectories would produce about another 120 KB of trajectory data and a 100-KB PDB file containing the refined structure information. Thus, for an I/O approach in which all input and output is done by the master thread, two I/O bottlenecks occur. The first occurs at initial startup of the run, and the second occurs during computation. At startup, the database must be read and distributed to all nodes. This consists of 65 MB of data that can be read from disk and cached on the master node. Assuming a crude broadcast, in which the master thread contacts each CPU in turn, then with 32,768 processors, the master node must broadcast 2 TB of data.

With a point-to-point speed of approximately 160 MB/s, this corresponds to more than 3 hours of elapsed time. Obviously, a binary tree approach could address this challenge; however, it is equally efficacious to simply have each worker thread read the database from disk via the parallel file system.

During the actual computation, the size of a work packet (containing data for a single trajectory) sent from the master node to each worker node would typically be on the order of 100 KB. The returned completed work packet size would be on the order of 220 KB. If the calculation of 120,000 trajectories was to be completed in 3 hours, then this corresponds to a sustained I/O load on the master thread of 3.5 MB/s, which, though technically possible, assumes that the datastream is continuous or that sufficient buffer space is available to fully buffer the I/O stream. On the BG/L system in particular, this initial approach caused serious problems with memory management since it not only requires excessive memory for buffering the output but it also requires a significant amount of memory on the master node to deal with the multiple MPI events. This meant that Rosetta structure predictions were initially restricted to about 80 amino-acid residues on BG/L systems. By transitioning to a distributed I/O model in which each worker thread writes to its own output file and both reads and writes its own structure files, we have alleviated most of the I/O constraints. In this case, the master thread has to send only minor job-control information over the MPI network and all I/O is distributed evenly across the parallel General Parallel File System (GPFS). When we move forward to greater than 100,000 processors for studying larger proteins and use somewhat more sampling, we may have to readdress this problem in order to avoid issues related to having extremely large numbers of individual files in a single directory, which, in our experience, not only makes post-analysis and archiving of the results difficult but also increases the I/O latency. This challenge might be addressed with a hierarchical directory structure or alternatively via some kind of connection to a back-end database rather than a flat file system. Additionally, we plan to implement a better monitoring and fault tolerance system so that the user can more easily monitor a calculation when it is running on an extremely large number of nodes.

Another problem that was first encountered when moving to massively parallel machines was the way in which Rosetta treats random numbers. By default, Rosetta seeds the random number generator using the time of day. On a single system, this ensures a unique seed and so avoids any duplication in the production run. For testing purposes, the user can override this with his own seed. When we initially implemented the MPI version of the code, we simply let each thread obtain its random number seed from the time of day. Delays in the startup of each thread meant that for simulations using up to approximately 128 processors, each thread would get a different initial seed. However, upon moving beyond 128 processors, it quickly became evident that multiple threads could obtain the same random number seed. This would then lead to the two threads duplicating the same result and so having a negative impact on the efficiency. The solution we have chosen for this is to have only the master thread obtain the time of day and then broadcast this to each of the worker threads at the commencement of the run. Each worker thread then multiplies its task ID by the time-of-day value and uses this as the random number seed. This approach avoids any issues with duplication and also allows for testing on large numbers of CPUs since the user can specify the random seed and each worker thread will inherit this and multiply its own task ID by the user-defined number. Obviously, this means that for exact duplication of results, the user must run the simulation on exactly the same number of processors, but for the purposes of testing the software, this is sufficient. Ideally, one would like to be able to exactly reproduce the same ensemble of trajectories, but this will require considerable reworking of the code since the master thread will have to send specific blocks of random number sequences to each worker thread in a way that is independent of the number of threads used or the order of operation. We have plans to implement this to facilitate easier testing in parallel, but for now, our current implementation is perfectly acceptable in terms of the scientific results generated.

Overall, the implementation of a master/worker job distribution system, coupled with parallel I/O, has been sufficient to obtain efficient scaling performance to 40,960 processors on the BG/L system, which is the largest calculation (in terms of total numbers of processors) that we have run to date. The scaling behavior for an 86-residue protein is highlighted in **Figure 2**. On 40,960 processors, we observe approximately 86% efficiency. The 14% loss of efficiency is largely due to a combination of load-balancing issues at high processor counts and increasing disk latency caused by contention for the I/O nodes. We are hopeful that future modifications aimed at scaling to more than 100,000 processors, as discussed later, will address this problem and further improve the scaling efficiency.

Results from CASP7 study

The phrase “Critical Assessment of Techniques for Protein Structure Prediction” (CASP) [16, 17] refers to a biennial community-wide experiment to test the current state of the art in computational methods for protein structure prediction. The CASP organizers release the primary sequence of a number of proteins for which the experimental structures are not currently available to the scientific community. Each team then conducts blind predictions of the 3D protein structures and submits them to the CASP organizers. At the end of the experiment, the experimental structures are made public and the predictions of each team are scored against these previously unknown experimental structures. CASP, therefore, acts as an important benchmark in measuring the accuracy of current protein structure prediction algorithms and is considered very seriously throughout the protein structure prediction community.

There is typically a 3-week window between the release of the sequence and the due date for submission of the prediction. In CASP7, about 100 targets were made available for prediction from May through July 2006, corresponding to nearly a target a day. Each of these targets required a large volume of computing before a reliable prediction, using Rosetta, could be made. Ideally, one would like to adopt the approach of using one large production run, followed by significant post-production analysis, for each target. However, in previous CASPs, we were restricted by the lack of available computing resources and had to use a less efficient approach that attempted to iteratively refine the structure. Furthermore, with a large fraction of time spent on computing, typically very little time remained for thorough post-production analysis of the models. The post-production analysis is as important as generating the models themselves. CASP7 marked the first time in which we had access both to a version of Rosetta that could run on massively parallel systems and to machines in which a large amount of computing time was available for a reasonable length of time (e.g., 24 hours). In the latest CASP experiment, we made extensive use of the BG/L systems at the San Diego Supercomputer Center and Argonne National Laboratory. Overall, the access to tightly integrated and reliable massively parallel systems allowed us to run our prediction algorithm for a significantly longer duration of simulation time, allowing a more thorough exploration of conformation space and affording greater time for analysis. A complete overview of the CASP7 performance of the Rosetta team is discussed elsewhere [19]. An overview of the performance of all the groups that took part in CASP7 can also be obtained from the CASP Web site [16]. Below, we discuss a specific example that we ran as a demonstration of the effectiveness of massively parallel systems for protein structure prediction.

Running Rosetta on 16 racks of the BG/L system for CASP7

In order to demonstrate the utility of massively parallel supercomputers when coupled with recent modifications to the Rosetta software, we participated in the July 2006 IBM BG/L application deployment workshop hosted on the 20-rack BG/L system at the IBM T. J. Watson Research Center (BGW). This coincided with the CASP7 competition and so provided us with an opportunity to test the potential of the Rosetta software to effectively utilize a massively parallel (>100-Tflops) supercomputer on a current CASP target. The organizers of the workshop made 16 racks (32,768 CPUs) of the BGW supercomputer facility available to us for 3 hours. Our aim was to be able to run a full Rosetta production run on a moderate-size CASP target, leaving ample time for thorough analysis after the production run.

We selected CASP target T0380. The experimentally determined structure of this protein is now published in the Brookhaven PDB of experimentally determined protein structures as PDB ID 2HQ7 [20] and is the general stress protein 26 (GS26) of *Bacillus subtilis* (pyridoxine-phosphate oxidase family). This target fell into the template-based modeling category of the CASP experiment. When the primary amino-acid sequence of T0380 was compared against sequences of all structures deposited in the PDB, we found matches to structures that shared significant sequence similarity with the query

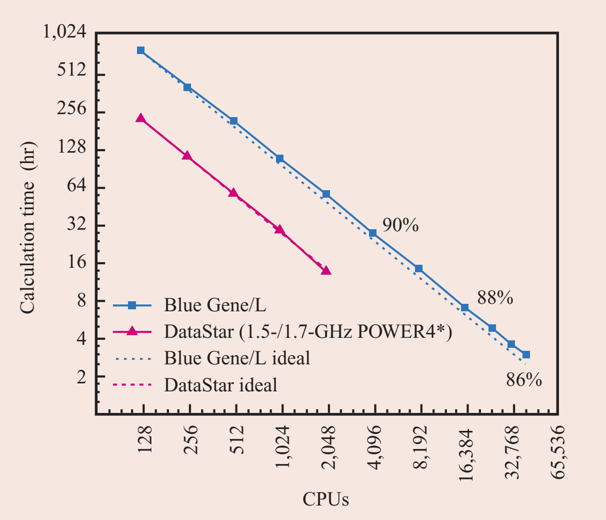

Figure 2

Rosetta scaling on a BG/L system and a DataStar system as a function of CPU count. DataStar is an IBM POWER4 (Federation interconnect) system located at the San Diego Supercomputer Center. Calculation time is the time to complete 120,000 all-atom refinements of an 86-residue protein. With the exception of the 32,768 CPU run, the timings were calculated from benchmark runs on a subset of the 120,000 refinements and then extrapolated. The ideal line represents the time taken to run all the refinements on one processor divided by the number of processors. Note that the DataStar runs made use of a combination of 1.5-GHz and 1.7-GHz nodes, depending on the number of CPUs in use.

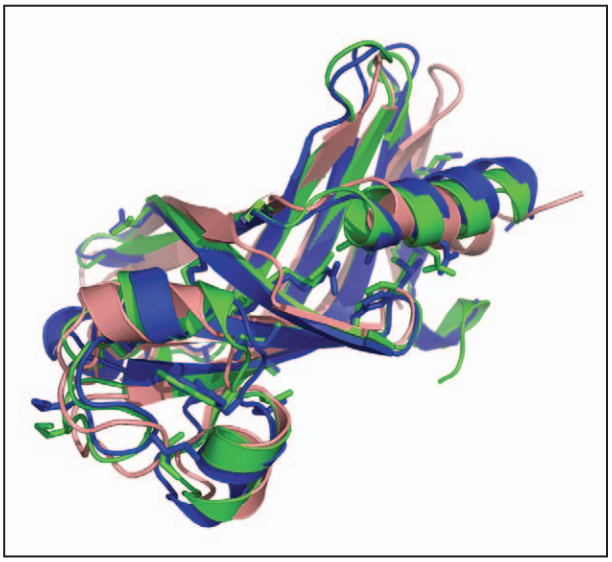

Figure 3

Superposition of the native structure (which was made publicly available after the prediction was submitted for review) in blue, the initial template in pink, and the Rosetta prediction in green for CASP7 target T0380. All production calculation for this target was completed in fewer than 3 hours on 32,768 processors of the BGW system.

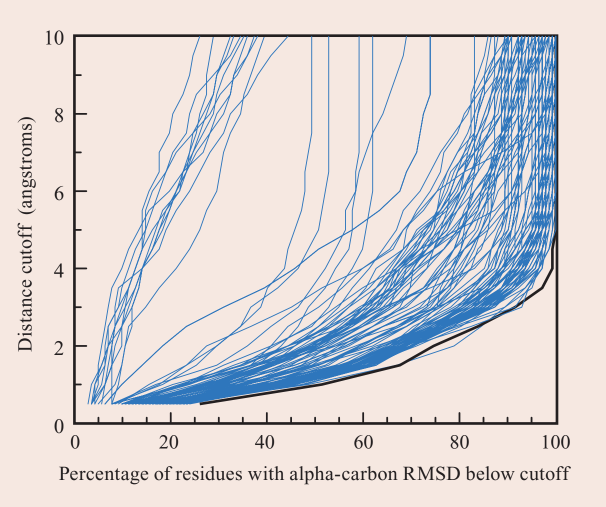

Figure 4

Plot showing the percentage of residues that had alpha-carbon Cartesian root-mean-square deviations (RMSDs) between the predicted structure and the experimental structure, below a distance cutoff in angstroms (10^{-10} m) shown on the vertical axis. The closer the line is positioned to the bottom right corner of the plot, the better the fit. The first submitted prediction for each of the teams that predicted target T0380 as part of CASP7 is shown. The Baker research group prediction is shown in black.

sequence. It was likely that these structures, slightly varying between themselves, resembled the unknown T0380 structure. Therefore, we used the structure with the highest sequence identity as a template, or starting point, for modeling T0380. The T0380 sequence was threaded onto the template and Rosetta refinement was applied. We then performed a single Rosetta production run on the system for 3 hours, utilizing a total of 98,304 CPU hours. The calculation generated approximately 190,000 models. We selected the lowest-scoring 5% for further processing and analysis. The post-production analysis, routinely used, includes distance-based hierarchical clusters, filtering out models for poorly packed regions of the protein, ensuring among other things that buried polar residues are involved in hydrogen bonding. A second smaller round of simulation was performed on desktop compute resources to ensure that the desired features were present in the model. Access to the BGW system allowed us to complete the entire prediction within a day and quickly submit it to the CASP organizers. **Figure 3** shows the superposition of the native structure (released later) in blue, template structure in pink, and the Rosetta prediction in green. **Figure 4** shows the results for the first prediction submitted by each of the teams that predicted this target. This plot shows the percentage of residues (alpha-carbons) in the predicted structure that had root-mean-square deviations (RMSDs) to the crystal structure of less than the cutoff shown on the *y*-axis. Thus, lines that have a low initial gradient and a low *y*-axis value indicate better fits. The prediction from the David Baker research group is highlighted in black, and it is clear that this was one of the most accurate predictions.

Future development paths aimed at next-generation HPC systems

Since the current development path for HPC systems is targeted toward larger and larger degrees of parallelism, it is essential that we attempt to address the issues of running Rosetta with more than 100,000 processors. As mentioned earlier, a number of problems are encountered when running Rosetta on such a large number of processors, and we plan a number of modifications to help improve the efficiency.

First, the single master thread system will be incapable of handling such a large number of worker threads; thus, we plan to investigate hierarchical approaches for job distribution. This should reduce contention for the master node. Second, beyond 100,000 processors, it becomes necessary to start actively dealing with node failures, which are likely to be fairly frequent (<24 hours per failure). To address this, we are looking at ways to actively monitor nodes. One potential option is for the master thread, or the group master in a hierarchical system, to routinely check on each of the worker threads.

Since the master thread keeps a record of what each worker is working on, if a worker is found to be non-responsive, the tasks assigned to that worker thread can be distributed to other functional threads as necessary.

We also must attempt to improve issues with I/O latency when using massively parallel systems. One potential approach is to attempt to buffer the I/O locally on a node and then write the buffer out as larger chunks in less frequent intervals. This will partially alleviate the problem, but since the memory-to-processor ratio of most future HPC systems is likely to be constant, or even decreasing, this may yield only a short-term fix. Ultimately, we plan to overhaul the software in a way that allows better management of the file structure and job progression. One potential option here is to make use of either a hierarchical or a database-like file structure as opposed to the flat file system we currently use. This will obviously make analysis of the results easier, but it will also likely improve the I/O latency. In recent simulations, we have noticed that the latency associated with writing an individual output file to disk, in the form of a protein databank file containing the coordinates of a predicted structure, can increase as the number of files within a specific directory grows. We hope that avoiding flat file systems will improve this situation.

Finally, as we move toward machines with processor counts approaching one million, it will be necessary to migrate to an approach whereby the worker threads represent more than a single processor. Without this, load balancing will be a serious issue, and it will not be possible to effectively utilize the entire machine. The restricted memory per processor will also prevent simulations of large sequences (>120 amino acids) from being run. For reference, **Figure 5** shows a histogram of the sequence lengths (in amino-acid residues) of all current entries in the PDB. This clearly shows that the Rosetta software might one day be used for much larger proteins. Since our ultimate goal is to run simulations that involve roughly constant numbers of refinements (~250,000) but with larger and larger counts of amino acids, it will become necessary to implement a finer degree of parallelism within the refinement step. This is necessary both to reduce the wall-clock (elapsed) time per individual refinement and also to make the effective memory available per MPI-refinement thread larger than the current 256 MB provided by the BG/L system. One possible approach may include a hybrid OpenMP**/MPI (or custom threaded) approach whereby we maintain the communications between job distributors and workers using MPI, but within a worker thread we utilize OpenMP or custom threading to, for example, run 32 SMP (symmetric multiprocessing) threads for a single refinement calculation. While technically fairly simple to implement, this approach will restrict the software to HPC systems that are built out of reasonably large SMP blocks. An alternative would be to implement a subset of MPI communicators and perform the refinement in parallel over a subset of processors using MPI. Though more portable, such an approach will lead to a significantly more complicated code base and also does not improve the restrictions imposed by limited memory per processor. Thus, in order to achieve the goal of both structure prediction and functional studies of more and more complex proteins, it will be essential that future massively parallel systems support large (>16 processors) SMP nodes with a memory capacity that is not less than 256 MB per processor (for a minimum of 4 GB fully shared and addressable per node). Without this, the memory limitations imposed by the desire to achieve higher and higher peak floating-point-operation counts at the expense of system balance will undoubtedly restrict future advances in this field.

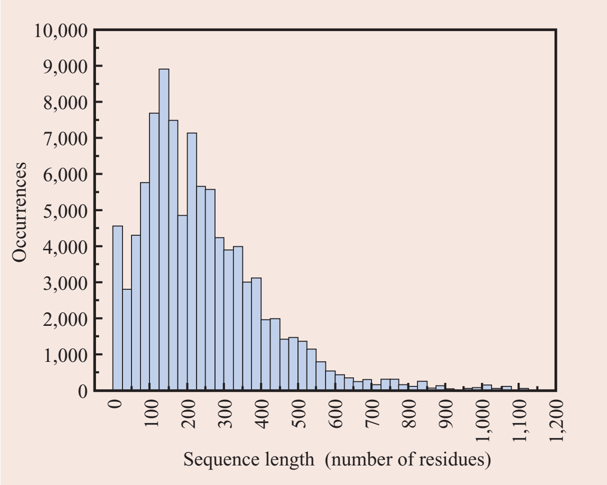

Figure 5

Histogram showing the sequence length (number of amino-acid residues) for all protein structures currently deposited in the Protein Data Bank. This information was compiled from information at the Protein Data Bank [21].

Conclusions

CASP7 marks the first time that we have had access to both massively parallel high-performance computational resources and a version of the Rosetta software that could take advantage of them. The ability to run individual protein structure predictions in a relatively short amount of time (<1 day) while dramatically increasing the amount of sampling per prediction made a significant impact on our approach in CASP7 and will undoubtedly influence the way in which we approach CASP8.

In addition, access to massively parallel high-performance computer systems has been extremely beneficial for protein structure prediction using Rosetta by greatly reducing the time required to perform a single prediction and so increasing the size of proteins that can be studied. When we use ever-more-powerful machines in the future, we face a number of hurdles that must be overcome to make efficient use of these machines. However, we are confident that none of these challenges is insurmountable, and it is highly likely that future advances in massively parallel computer systems will increase protein structure prediction throughput and, more importantly, lead to refinements of the underlying models and to more accurate protein structure prediction.

Acknowledgments

We acknowledge support from the Howard Hughes Medical Institute, the American Leukemia and Lymphoma Society, and the National Science Foundation. We also thank the San Diego Supercomputer Center for its continued support of this project and future developments through its Strategic Applications Collaboration program. We also thank IBM for its generous donation of supercomputing time on the BGW platform.

*Trademark, service mark, or registered trademark of International Business Machines Corporation in the United States, other countries, or both.

**Trademark, service mark, or registered trademark of the OpenMP Architecture Review Board in the United States, other countries, or both.

References

1. C. B. Anfinsen, E. Haber, M. Sela, and F. H. White, Jr., "The Kinetics of Formation of Native Ribonuclease During Oxidation of the Reduced Polypeptide Chain," *Proc. Natl. Acad. Sci. USA* **47**, No. 9, 1309–1314 (1961).
2. D. Baker and A. Sali, "Protein Structure Prediction and Structural Genomics," *Science* **294**, No. 5540, 93–96 (2001).
3. C. Levinthal, "Are There Pathways for Protein Folding?" *J. Chim. Phys. Pys.-Chim. Bio.* **65**, No. 1, 44–68 (1968).
4. K. T. Simons, C. Kooperberg, E. Huang, and D. Baker, "Assembly of Protein Tertiary Structures from Fragments with Similar Local Sequences Using Simulated Annealing and Bayesian Scoring Functions," *J. Mol. Biol.* **268**, No. 1, 209–225 (1997).
5. R. L. Dunbrack, Jr. and F. E. Cohen, "Bayesian Statistical Analysis of Protein Side-Chain Rotamer Preferences," *Protein Sci.* **6**, No. 8, 1661–1681 (1997).
6. B. Kuhlman and D. Baker, "Native Protein Sequences Are Close to Optimal for Their Structures," *Proc. Natl. Acad. Sci. USA* **97**, No. 19, 10383–10388 (2000).
7. K. M. Misura and D. Baker, "Progress and Challenges in High-Resolution Refinement of Protein Structure Models," *Proteins* **59**, No. 1, 15–29 (2005).
8. P. Bradley, K. M. Misura, and D. Baker, "Toward High-Resolution *De Novo* Structure Prediction for Small Proteins," *Science* **309**, No. 5742, 1868–1871 (2005).
9. Z. Li and H. A. Scheraga, "Monte-Carlo Minimization Approach to the Multiple Minima Problem in Protein Folding," *Proc. Natl. Acad. Sci. USA* **84**, No. 19, 6611–6615 (1987).
10. T. Lazaridis and M. Karplus, "Effective Energy Function for Proteins in Solution," *Proteins* **35**, No. 2, 133–152 (1999).
11. A. V. Morozov, T. Kortemme, K. Tsemekhman, and D. Baker, "Close Agreement Between the Orientation Dependence of Hydrogen Bonds Observed in Protein Structures and Quantum Mechanical Calculations," *Proc. Natl. Acad. Sci. USA* **101**, No. 18, 6946–6951 (2004).
12. T. Kortemme, A. V. Morozov, and D. Baker, "An Orientation-Dependent Hydrogen Bonding Potential Improves Prediction of Specificity and Structure for Proteins and Protein–Protein Complexes," *J. Mol. Biol.* **326**, No. 4, 1239–1259 (2003).
13. A. A. Canutescu and R. L. Dunbrack, Jr., "Cyclic Coordinate Descent: A Robotics Algorithm for Protein Loop Closure," *Protein Sci.* **12**, No. 5, 963–972 (2003).
14. B. Kuhlman, G. Dantas, G. C. Ireton, G. Varani, B. L. Stoddard, and D. Baker, "Design of a Novel Globular Protein Fold with Atomic-Level Accuracy," *Science* **302**, No. 5649, 1364–1368 (2003).
15. C. A. Rohl, C. E. Strauss, K. M. Misura, and D. Baker, "Protein Structure Prediction Using Rosetta," *Methods Enzymol.* **383**, No. 1, 66–93 (2004).
16. Protein Structure Prediction Center, Genome Center, University of California, Davis; see *http://predictioncenter.org/*.
17. J. Moult, K. Fidelis, A. Zemla, and T. Hubbard, "Critical Assessment of Methods of Protein Structure Prediction (CASP)-Round V," *Proteins Str. Func. Gen.* **53**, No. 6, 334–339 (2003).
18. M. Snir, S. W. Otto, S. Huss-Lederman, D. W. Walker, and J. Dongarra, *MPI: The Complete Reference*, Cambridge, Massachusetts, London, England: The MIT Press, Cambridge, MA, London, England, 1996.
19. R. Das, B. Qian, S. Raman, R. Vernon, J. Thompson, P. Bradley, S. Khare, et al., "Structure Prediction for CASP7 Targets Using Extensive All-Atom Refinement with Rosetta@home," *Proteins Structure Function Bioinformatics* **69**, No. S8, 118–128 (2007).
20. I. A. Wilson, for the Joint Center for Structural Genomics, Structure 2HQ7, 2007; see *http://www.pdb.org/pdb/explore/explore.do?structureId=2HQ7*.
21. "Research Collaboratory for Structural Bioinformatics (RCSB)" (sequence data); see *ftp://ftp.rcsb.org/pub/pdb/derived_data/pdb_seqres.txt*.

Received March 17, 2007; accepted for publication April 11, 2007; Internet publication December 5, 2007

Srivatsan Raman *Department of Biochemistry, University of Washington, J Wing Health Sciences Building, Box 357350, Seattle, Washington 98195 (sraman@u.washington.edu)*. Mr. Raman is a graduate student in Professor David Baker's research group at the University of Washington. His Ph.D. thesis work is in the field of high-resolution protein structure prediction. He uses Blue Gene/L supercomputers at the IBM T. J. Watson Research Center, the San Diego Supercomputer Center, and Argonne National Laboratory for running protein structure prediction calculations.

Bin Qian *Department of Biochemistry, University of Washington, J Wing Health Sciences Building, Box 357350, Seattle, Washington 98195 (bqian@u.washington.edu)*. After receiving his Ph.D. degree in biophysics from the University of Michigan, Dr. Qian became a Postdoctoral Fellow at the University of Washington, working with Professor David Baker, with support from the Leukemia and Lymphoma Society career development fellowship. Dr. Qian's main expertise is protein structure and function prediction.

David Baker *University of Washington, J Wing Health Sciences Building, Box 357350, Seattle, Washington 98195 (dabaker@u.washington.edu)*. Professor Baker is a Principal Investigator for the Howard Hughes Medical Institute. He is the recipient of Young Investigator Awards from the Packard Foundation, the National Science Foundation, and the Beckman Foundation, and he is the recipient of the Irving Sigal Young Investigator Award from the Protein Society and the Overton Award from the International Society of Computational Biology. He is considered one of the leading experts in protein structure prediction and enzyme design, and since 1992, he has published more than 140 papers in internationally renowned peer-reviewed journals, including five in *Science* and *Nature*. He is the principal author of the protein structure prediction software Rosetta.

Ross C. Walker *San Diego Supercomputer Center, University of California at San Diego, 9500 Gilman Drive #0505, La Jolla, California, 92093-0505 (rcw@sdsc.edu)*. Dr. Walker received his Ph.D. degree in computational chemistry from Imperial College London and then worked as a National Institutes of Health-funded Postdoctoral Fellow at The Scripps Research Institute (TSRI) in the research group of Professor David Case. While at TSRI, he worked as a principal author of the AMBER molecular dynamics software incorporating a completely new QM/MM molecular dynamics module. In March 2006, he moved to an Assistant Research Professor position at the San Diego Supercomputer Center where he works in the Scientific Computing Applications Group. Through the National Science Foundation-funded Strategic Applications Collaboration program, Dr. Walker specializes in maximizing the potential impact of high-performance computer systems on molecular biology research.