

A Parallel Algorithm for Calculating the Potential Energy in DNA

John S. Conery,^{*} Warner L. Peticolas,[†] Thomas Rush III,[†]
Kesavan Shanmugam,^{*} and Jose Dominguez^{*}

^{*} Department of Computer and Information Science

[†] Department of Chemistry

University of Oregon

Eugene, OR 97403

email: conery@cs.uoregon.edu

Abstract

The Dreiding force field is a seven-term equation that describes the potential energy in a molecule as a function of the relative positions of bonded atoms and electrostatic interactions between atoms that do not share a bond. For large molecules such as DNA, with several thousand atoms, the $O(n^2)$ nonbonded terms can require a significant amount of computation. In this paper we present a data-parallel algorithm that takes time $O(n)$ on n processors. We compare the execution time of our algorithm on a MasPar MP-1 with an efficient sequential program running on an SGI workstation.

1 Introduction

The long polymeric structure of deoxyribonucleic acid (DNA) may change significantly as it performs its vital functions in living cellular organisms. The two main functions of DNA include replication, which is needed for growth and the passing along of hereditary information, and the transcription of the ribonucleic acids which make the proteins that regulate most of our biological processes. To help understand DNA's role in such events, researchers have been expending considerable effort in determining the three-dimensional structure, or *conformation*, of DNA under a great variety of conditions.

In the past, X-ray crystallography has been a tremendous help in determining some of the conformations that DNA may adopt. It has the advantage that it experimentally provides the average Cartesian coordinates for every atom in a molecule, and therefore the overall conformation, but it also has several disadvantages when it comes to studying biological molecules in their natural environment. Two major disadvantages are that good crystalline DNA sam-

ples are not always easy to obtain, and that DNA is not naturally found as a solid.

A promising approach to this problem is to combine computer modeling with experimental methods to provide a better description of the structure of DNA. This approach uses an experimental technique such as Nuclear Magnetic Resonance (NMR) or Raman spectroscopy to provide an initial conformation and then uses a minimization algorithm to find the lowest energy conformation in the neighborhood of the starting point (e.g [2]).

In this paper we describe a new data-parallel algorithm for calculating the energy of a single conformation of a DNA molecule. Our algorithm will compute the potential energy in a molecule with n atoms, taking into account the $O(n^2)$ interactions between all the atoms. The parallel algorithm runs in time $O(n)$ on n processors, which is an improvement over the $O(n^2)$ time complexity of a serial algorithm running on one processor.

In order to avoid calculating all n^2 interactions, it is common to set a "cutoff" distance beyond which energies are not calculated; for example, by setting the cutoff to 10 angstroms (\AA) the program will not compute the coulombic attraction of any pair of atoms further than 10\AA apart. Since our algorithm is linear in the number of atoms it is easier to calculate all interactions and thus achieve a more accurate value of the total energy. The difference in the calculated energy is nontrivial — 15% in one example molecule containing 1264 atoms — and thus it is worthwhile to compute all interactions when possible. In Section 6 we give a table that shows the energy values with and without cutoffs. We also compare the execution times of a serial algorithm running on a Silicon Graphics Indigo workstation (part of a commercial package named BIOGRAF [1]) and our parallel algorithm running on a MasPar MP-1.

2 Background: Raman Spectroscopy

Over the past twenty years, Raman spectroscopy has been developed to identify particular conformations of oligonucleotides (short synthetic strands of DNA) in solution, crystal and fiber forms [7]. Only recently has this information been able to be used to determine a reasonable set of Cartesian coordinates for a whole oligomer of DNA in solution. To date, only a few solution structures have been solved because without other supporting experimental data (such as NMR data) there are a large number of calculations which need to be performed.

Raman spectroscopy provides information about the energy (frequency) of the vibrations within a molecule [5]. Data are obtained by shining a monochromatic laser on the sample and detecting the light being scattered. A monochromator separates the light into its component frequencies, and a photomultiplier tube detects the intensity of light of the different frequencies. Different vibrations within a molecule will produce different peaks at different frequencies. The most likely structure for the oligomer in solution is the one which will reproduce the Raman spectrum and have the lowest energy.

It is also true, and very important for this application, that a particular vibration will have a slightly different frequency depending on the geometry of the molecule. An example of this is the Guanine breathing vibration, which appears at 625 cm^{-1} , 665 cm^{-1} , and 680 cm^{-1} depending on three different conformations of the DNA molecule (the unit cm^{-1} is a wavenumber, which is directly proportional to frequency).

Over the past twenty years, researchers have compared Raman spectra of crystals with known Cartesian coordinates to make a list of Raman conformational markers (some theoretical calculations were also performed). Using these markers, a Raman experiment can determine most of the structural information of a molecule with unknown geometry. This rough guess at a geometry may then be entered into an energy minimization program where the experimental geometry is to be determined.

The Raman data can potentially dictate a structure which has torsional angles (see Section 3.3) which fall within ± 10 degrees of the average solution conformation of the molecule. With this initial guess at the structure and a well defined, constrained region of the molecule's multidimensional conformational space, it is possible to more easily search for the locally global minima.

The long range goal of our research is to provide a more accurate description of the average conformation of large biological molecules by combining experimental and computational methods. Neither class of methods alone is accurate enough. X-ray crystallography analyzes DNA and other molecules in crystalline form, which has inherent

inaccuracies, and Raman and NMR methods can give only partial information at well-known markers. Computational techniques alone are also inadequate: minimization algorithms are prone to becoming trapped in local minima and are very expensive for larger molecules since there are so many dimensions in the search space. A combined experimental and computational method may be able to overcome these difficulties and provide a more accurate picture than either method alone.

The combined Raman/minimization technique will consist of

1. using Raman conformational markers to obtain an initial guess at a structure;
2. minimizing this structure to obtain a refined initial structure;
3. systematically minimizing many initial conformations within the conformational space described by the vibrational spectra; and finally
4. analyzing these minimized structures to obtain a final average solution structure.

3 Energy Calculations

The energy calculations are performed using the Dreiding force field [6], which is the sum of seven independent components:

bond energy:

$$\sum_b \frac{1}{2} k_b (R - R_0)^2$$

angle energy:

$$\sum_\theta \frac{1}{2} k_\theta (\theta - \theta_0)^2$$

torsion energy:

$$\sum_\tau \sum_{n=1}^p \frac{1}{2} k_{\tau, n} [1 - d \cos(n\phi)]$$

inversion energy:

$$\sum_i \frac{1}{2} C (\cos \omega - \cos \omega_0)^2$$

Coulombic attraction:

$$\sum_{R_{ij}} C_0 \frac{Q_i Q_j}{\epsilon_0 R_{ij}^2}$$

van der Waals interaction:

$$\sum_{R_{ij}} D_0 \left[\left(\frac{R_0}{R_{ij}} \right)^{12} - 2 \left(\frac{R_0}{R_{ij}} \right)^6 \right]$$

hydrogen bond interaction:

$$\sum_{R_{ij}} D_0 \left[5 \left(\frac{R_0}{R_{ij}} \right)^{12} - 6 \left(\frac{R_0}{R_{ij}} \right)^{10} \right] \cos^4 \theta_{iHj}$$

The first four terms in this energy field are the *bonded* terms. They represent contributions to the total energy arising from atoms which are directly bonded to each other. The four situations which are taken into account here are:

1. the energy due to a bond being stretched or compressed;
2. the energy due to an angle formed by three atoms being displaced from its equilibrium position;
3. the energy due to the torsional angle formed by four sequentially bonded atoms being displaced from its equilibrium position; and
4. the energy due to an atom being displaced from its equilibrium position above the plane of three nearby atoms (known as an inversion).

The remaining *nonbonded* terms are the electrostatic interaction energies of an atom with another atom in the molecule or the surrounding environment. In these calculations we do not compute nonbonded energy terms for atoms that are bonded, i.e. the electrostatic interactions between two bonded atoms is accounted for in the bond energy. In a large molecule, with a thousand or more atoms, calculating the nonbonded terms is the most time consuming part of the calculation since there are on the order of n^2 interactions in a molecule of n atoms.

The overall goal of the computation is to minimize this energy term by varying the conformation using an optimization algorithm such as a conjugate gradient or Newton method. In this paper we present an efficient parallel algorithm for computing the energy of a single conformation; at the end of the paper we describe how our algorithm can be embedded in an optimization algorithm.

3.1 Bonds

In the four bonded interactions — bond, angle, torsion, and inversion energies — a contribution to the total energy arises when atoms are displaced from their equilibrium

positions. In the bond term,

$$\frac{1}{2} k_b (R - R_o)^2$$

R_0 is the equilibrium bond distance specific for the pair of atoms in the bond, R is the current bond distance, which is calculated from their Cartesian coordinates, and k_b is a constant which is also specific for the pair of atoms. When R is different from R_o (either a positive or negative difference) this term contributes to the total energy. This is analogous to, and actually modelled after, two balls connected by a spring (Figure 1a). When the spring is stretched out past its natural length, it reacts by trying to return to its equilibrium position because its energy is too high. The same is true when you compress the spring, i.e. the bond is too short.

3.2 Angles

The term for the energy due to the angle formed by three atoms is identical to the bond term, except instead of current and equilibrium bond length we compare the equilibrium angle θ_0 to the current angle θ (Figure 1b). k_θ is again a constant that is specific for the three atoms in the angle. As with the bond term, only displacements from the equilibrium angle cause an energy contribution.

3.3 Torsions

Torsion refers to the force required to “twist” a bond about its horizontal axis. A torsional term will arise when there are four atoms i , j , k , and l linked by bonds i - j , j - k , and k - l . The current torsional angle ϕ is the angle between the jkl plane and the ijk plane (Figure 1c). The torsional term has the same concept as the bond and angle terms but is written slightly differently because there may be several equilibrium torsional angles at periodic intervals. The parameters that are specific to the four-atom combination making up the torsional angle are the periodicity p of the potential, the force constant $k_{\tau, n}$, and the locations of the p minima, which are accounted for by the constant d shown in the formula. The convention is to define a positive angle as clockwise looking from atom j toward atom k . In the Dreiding force field, it is also customary to rescale the force constant by dividing it by the number of torsions defined around that same bond in the molecule.

3.4 Inversions

An inversion occurs when an atom l that is at equilibrium with respect to three other atoms i , j , and k is pushed to the other side of the ijk plane (Figure 1d). The inversion

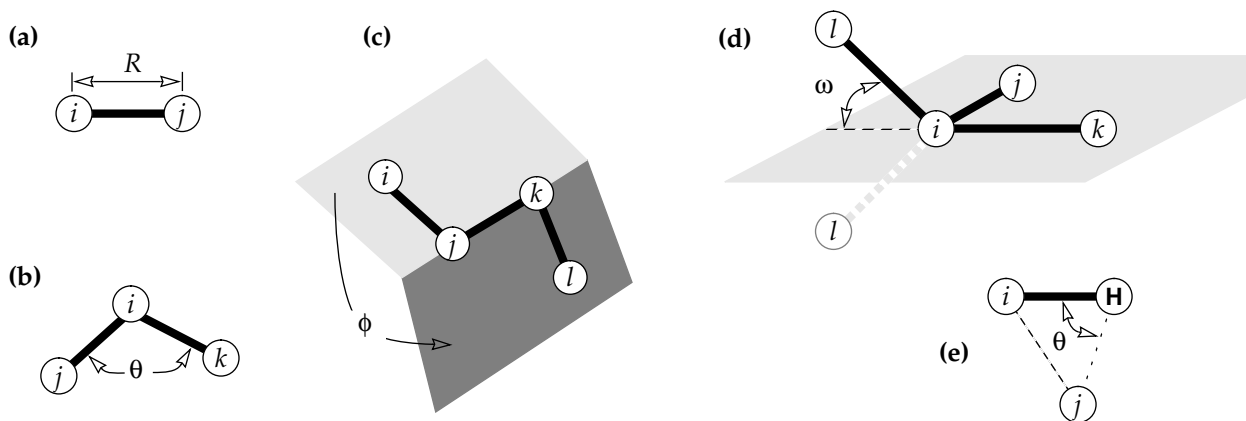


Figure 1. (a) A bond between atoms i and j that are R angstroms apart. (b) The angle θ formed by atoms i , j , and k ; atom i is the central atom. (c) Torsional energy is the result of “twisting” the bond between atoms j and k . (d) The inversion energy term, from moving atom l to the other side of the ijk plane. (e) Nonbonded hydrogen interaction between donor i and acceptor j ; i is bonded with a hydrogen, θ is the angle formed by iHj .

term has the same form as the bond and angle terms, where the inversion angle ω is the angle between the il axis and the ijk plane. The force constant here is labeled C , the current inversion angle is ω , and the equilibrium inversion angle is ω_0 .

3.5 Nonbonded Coulombic Interactions

In the coulombic term, the interaction is due to the charges Q_i and Q_j on atoms i and j , where like charges repel and unlike charges attract. This energy depends inversely on the distance between the atoms, which is again calculated from their Cartesian coordinates and scaled by $\epsilon_0 R_{ij}$. The scaling factor ϵ_0 is usually taken as 1, and the overall coulombic scaling factor, $C_0 = 332.0637$ puts the term in the proper units to be added in the total energy summation. As in all of the nonbonded interactions, the interactions between the atom being summed over and any atoms it is directly bonded to are not included. In the coulombic term we also ignore “1-3 interactions”, or the interaction between an atom and its second nearest bonded neighbor.

3.6 Nonbonded van der Waals Interactions

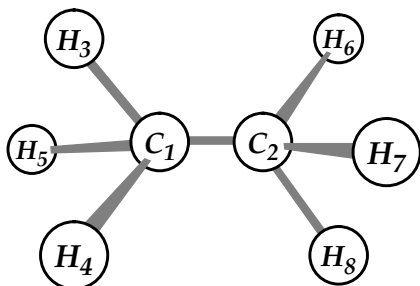
The van der Waals interaction is another type of electrostatic interaction. It has almost the same shape potential as the bond energy. The minimum energy is at the equilibrium van der Waals distance, R_0 , and the potential energy depends on the magnitude of the displacement from equilibrium. The constant D_0 is again specific to the atom pair involved. Due to limited experimental data, the R_0 and D_0 values between unlike atoms are calculated from the values for the interaction between like atoms. For example, R_0 for a C-H pair can be estimated from the known R_0 values for C-C and H-H interactions using the formula

$$R_0^{CH} = \sqrt{R_0^{CC} R_0^{HH}}$$

The values for D_0 are computed similarly, replacing R by D in the above formula.

3.7 Hydrogen Bond Interactions

The hydrogen bond interaction is calculated in the special case where a “donor” atom (either oxygen, nitrogen, or sulphur) has a bond to a hydrogen atom. The hydrogen’s single electron is weakly attached, and the donor has room in its outer shell for another electron, so the electron is pulled toward the donor. This alters the electrostatic charge between the donor and other atoms (known as acceptors).



| Bonds | Angles | | Torsions |
|-------|--------|-------|----------|
| 1-3 | 3-1-5 | 6-2-7 | 3-1-2-8 |
| 1-4 | 3-1-4 | 7-2-8 | 3-1-2-7 |
| 1-5 | 5-1-4 | 6-2-8 | 3-1-2-6 |
| 1-2 | 3-1-2 | 6-2-1 | 5-1-2-8 |
| 2-6 | 5-1-2 | 7-2-1 | 5-1-2-7 |
| 2-7 | 4-1-2 | 8-2-1 | 5-1-2-6 |
| 2-8 | | | 4-1-2-8 |
| | | | 4-1-2-7 |
| | | | 4-1-2-6 |

Figure 2. Ethane Molecule and its Bonds, Angles, and Torsions

The hydrogen bond term is a function of the angle iHj where i is the donor and j is the acceptor (Figure 1e). It is almost identical to the form of the van der Waals term except for the \cos^4 term.

4 Data Distribution

In order to calculate the energy of a configuration we need to distribute the data — both the Cartesian coordinates of the atoms and the relevant energy term constants — to the processing elements (PEs) that need the data. In our data-parallel approach we distribute the atoms among PEs so that each PE will hold the description of one atom and the bonds it participates in. A PE will perform all the calculations concerning its atom, which we call the “local atom.” In the description of the algorithm we will assume there are more PEs than atoms; the mapping of atoms to PEs when there are fewer actual PEs is straightforward and

can often be done by a compiler.

At the end of the data distribution, each PE will contain:

1. An `Atom` structure initialized with information about one atom
2. An array of bond information (one entry for each atom the local atom is connected to in the molecule)
3. Information about angles in which the local atom is the central atom
4. An array of torsions and inversions, with one entry for each possible torsion where the local atom is one of the two central atoms

The reason angles and torsions are limited to those in which the local atom is a central atom is to reduce the amount of redundant information. If we kept all the angles in which an atom participates every angle would be stored on three PEs, one for each atom in the angle.

Once the data has been distributed, the energy calculation problem becomes one of finding all the bond, angle, torsion and non-bonded terms in which this atom participates and then calculating the force due to each term. The total energy of the molecule is the sum of all these forces over all atoms, being careful not to include duplicate energy terms. For example, if there is a bond between atoms A and B , we want the processor for either A or B , but not both, to calculate the bond energy and contribute it to the final sum.

As an example, Figure 2 shows an ethane molecule and the bond, angle and torsion terms present in the molecule. The energy calculation algorithm should generate these terms and calculate the forces due to them. In addition, it should calculate the force due to non-bonded terms. The forces will be calculated in parallel on each PE and then combined (also in parallel) and forwarded to a front-end processor or user node for output.

4.1 The Visitor Algorithm

Our method for computing bond, angles, torsions, and inversions is based on the “Fox ring” algorithm used to compute the interactions in an n -body problem [8]. Our algorithm, which we dubbed the “visitor algorithm”, is similar to the Fox method in that descriptions of atoms are sent around a virtual ring to PEs where other atoms reside, but it has the extra complication that we need to compute angles, torsions, and other terms that involve more than two bodies.

If each PE has an atom, the visitation operation can be accomplished by shifting the atoms around a virtual ring, as illustrated in Figure 3. Almost every physical processor

topology has an embedded ring structure, so this communication pattern is likely to be very efficient [4][8]. It will take $n - 1$ steps to send an atom around the entire ring, where n is the number of atoms in the molecule.

We assume every PE maintains a local atom, which is named a in the following description. In addition, each PE holds the description of a second atom, named x . The step “send x to the right” means the PE should send the description of x on the communication channel that connects it to its neighbor on the right in the virtual ring. Similarly, “read x from the left” means the PE should read information from the communication channel connecting it to its left neighbor and store this description in x . Thus during the algorithm x is continually updated, and at any one time x holds the description of a visiting atom.

Visitor Algorithm (Bonds Only)

do in parallel on each PE that contains an atom a :

$x = a$

repeat $n - 1$ times:

send x to the right

read x from the left

if a is connected to x :

calculate the bond between a and x

else:

calculate the nonbonded interaction
between a and x

After $n - 1$ cycles an atom that starts at PE i will have passed completely around the ring and will now be visiting the PE directly to the left of PE i . During its journey atom i will be the visiting atom on every other PE in the ring, and each of these PEs will have had a chance to compute the energy term represented by their local atoms and the traveling atom i . Note also that during each cycle n PEs will simultaneously compute n different energy terms, corresponding to the interactions between their local atom and the atom that is currently visiting that PE.

Since there are $n - 1$ cycles in the outer loop of the visitors algorithm, the time required to compute all energy terms will be $O(n)$ on n processors, which is an improvement over the $O(n^2)$ that would be required on a single processor.

The above algorithm does not address the problem of duplicate calculations, and it omits the steps that do the calculations for angles and torsions. As we will see, communication will be more efficient if we go ahead and duplicate the bond calculations, but we will avoid duplicating angles and torsions.

4.2 Angles and Torsions in the Visitor Algorithm

The straightforward option for computing angles is to

allow PE_a to compute all the angles that contain atom a . This means that PE_a will generate angles of the form $a-x-x$, $x-a-x$, and $x-x-a$, where x is a nonlocal atom, and that every angle $a-b-c$ will be duplicated in PE_a , PE_b and PE_c .

A problem with this approach is that it requires a second pass around the ring because we do not know the order in which atoms will visit each other. For example, assume the molecule contains an angle $a-b-c$. With the straightforward approach the angle should be generated by PE_a . But if c visits PE_a before b does, this angle will not be generated because bond $a-b$ will not be present when c visits PE_a . Until b visits PE_a , PE_a will not know it has to look for c in order to compute the angle $a-b-c$. We can solve this problem by using two communication cycles, one for bonds and another for angles. On the first pass all the bonds are calculated, and by the second pass each PE will be able to detect when the third atom of an angle passes by.

A better approach simultaneously solves the problems of duplicate angle calculations and using an extra pass. Instead of generating all the angles that contain a , PE_a will generate only those angles in which a is the central atom. That is, PE_a will generate only angles of the form $x-a-y$. Now each angle occurs only once in the system — on the PE that is the home of the center atom — and two connecting atoms can be recognized and used on the first pass around the virtual ring.

Compute Angles

[done when atom x visits PE_a :]

if x is connected to a :

calculate and store bond $x-a$

for all bonds $a-b$ created so far:

calculate and store angle $x-a-b$

else:

calculate nonbonded interactions
between a and x

We can use a similar strategy for torsions and inversions. PE_a will compute torsions in which atom a occurs as either the second or third atom, i.e. torsions of the form $x-a-x-x$ or $x-x-a-x$. However, the previous problems have returned: each torsion is computed on two PEs, and it will take multiple passes to make sure a PE has all the information it needs. Once all the angles have been computed, however, it will take only one additional pass to compute the torsions. The complete visitor algorithm thus uses the first pass to compute bonds and angles and a second pass to compute torsions and inversions. We will correct for the fact that bonds, torsions, and inversions are computed twice during the summing phase of the algorithm, which is described next.

5 Summing the Energy Terms

When the visitor algorithm terminates, the bond, angle, torsion, and nonbonded energy terms are distributed across all the PEs. The final step is to collect and sum all the terms.

Every parallel processor has a method for applying a binary associative operator such as addition to elements that are distributed over all the PEs. This operation, often called a *reduction*, can be done in $O(\log_2 n)$ steps on n processors. For example, if a vector of 1000 elements is allocated such that x_i is stored on PE i , we can compute

$$\sum_{1 \leq i \leq 1000} x_i$$

in 10 steps ($\log_2 1000 \approx 10$).

In the following description, “ \sum bonds” means “the sum of all the bond energies for the atom a ,” i.e. all the bond forces computed at PE_a in the previous phases. “ \sum bonds” and “ \sum torsions” are angle and torsion forces computed at PE_a for atom a . The expression “reduce(x)” means “compute the sum of all the local variables x on each of the PEs, returning a single global sum.”

Sum Energy Terms

do in parallel on each PE that contains an atom a :

- f1 = \sum bonds
- f2 = \sum angles
- f3 = \sum torsions
- f4 = \sum inversions
- f5 = \sum nonbonded terms

bond-force = reduce(f1)/2
angle-force = reduce(f2)
torsion-force = reduce(f3)/2
inversion-force = reduce(f4)/2
nonbonded-force = reduce(f5)/2

$$E = \text{bond-force} + \text{angle-force} + \text{torsion-force} + \text{inversion-force} + \text{nonbonded-force}$$

Since the reductions are $O(\log n)$ the total time to compute the energy is $O(\log n + n) = O(n)$.

6 Implementation and Evaluation

We implemented the visitor algorithm on a MasPar 1104, an SIMD machine with 4096 processing elements. The code was written in MPL, a data-parallel language based on ANSI C. The virtual ring was implemented via the X-net, a nearest neighbor communication channel with toroidal wrap.

To make sure our program was producing the correct results and to compare our parallel algorithm with a very good serial algorithm we used BIOGRAF [1] to compute the free energy in seven molecules, ranging in size from 13 atoms to 2544 atoms. We ran BIOGRAF on a Silicon Graphics Indigo 1 workstation. We then computed the energy of the same seven molecules on the MasPar. The timing results are listed in Table 1.

The table shows two columns for BIOGRAF. The first column is the execution time when the nonbonded interactions were cut off at 10\AA , i.e. when two atoms were farther

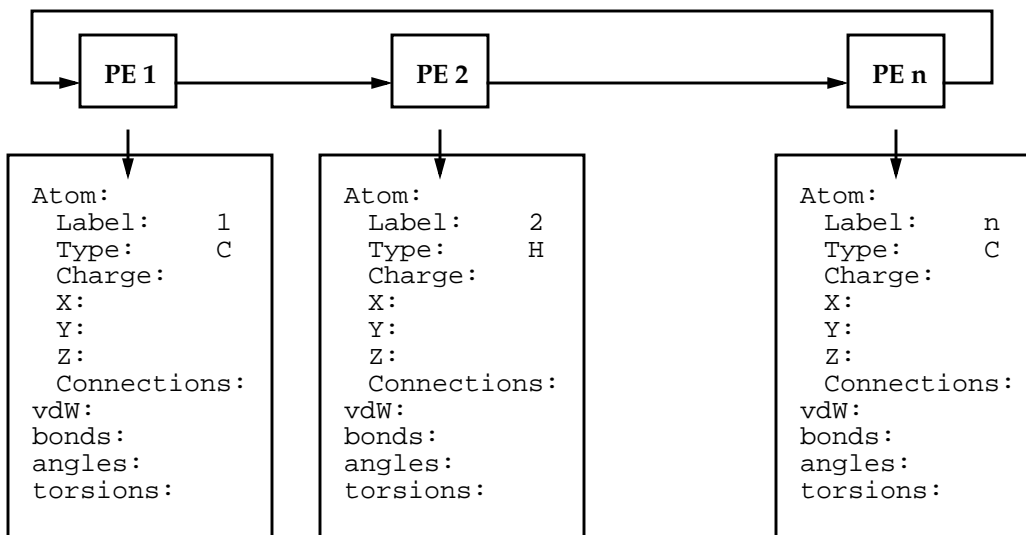


Figure 3. A Ring of Processors

Table 1: Sequential Execution on SGI vs. Parallel Execution on MasPar

| Molecule Size (#Atoms) | Number of Nonbonded Terms | Execution Time (seconds) | | | | Cutoff Error (%) |
|------------------------|---------------------------|--------------------------|-----------|-------------------------|------------------|------------------|
| | | BIOGRAF on SGI | | MasPar | | |
| | | cutoff | no cutoff | bonds, angles, torsions | calculate energy | |
| 13 | 52 | 0.05 | 0.07 | 0.14 | 0.45 | < .1 |
| 66 | 2,118 | 0.16 | 0.18 | 0.85 | 1.75 | < .1 |
| 135 | 8,838 | 0.25 | 0.31 | 1.35 | 4.07 | 5.8 |
| 294 | 43,179 | 0.31 | 0.84 | 3.37 | 5.37 | 4.0 |
| 964 | 471,812 | 1.41 | 55.37 | 9.69 | 15.63 | 7.7 |
| 1,264 | 824,992 | 2.74 | 471.47 | 12.55 | 20.26 | 15.1 |
| 2,544 | 3,273,773 | 7.46 | — | 25.07 | 39.85 | — |

than 10Å apart the program did not compute any non-bonded energy terms. The second column shows the execution time when all nonbonded terms were calculated, which was achieved by setting the cutoff distance to a very large value. The last column in the table shows the difference in energy (as a percentage) when it was calculated with and without all nonbonded interactions. From this small set of data it appears that the error grows with the size of the molecule, and that it will be important to use all interactions when searching for the minimal energy conformation of large molecules.

The execution times shown for our MasPar program are the time to construct the bonds, angles, and torsions (two passes around the virtual ring) and the time to calculate the energy of a single conformation (one pass, calculating all seven terms at each point). It does not include the time to load the descriptions of the atoms to individual PEs, which is substantial on our machine with a low bandwidth connection between the PEs and the front-end workstation (the time to load the descriptions of every atom in a molecule is comparable to the time to calculate the energy).

The execution times for BIOGRAF measure only the time to compute the energy. BIOGRAF allows users to specify which of the seven energy terms to use in a calculation. We first ran the program and measure the execution time t_0 when no energy terms were included. We then ran the program again, with and without nonbonded interactions, and subtracted t_0 from these times to get the times shown in the table. With 2544 atoms there were too many hydrogen bond interactions and BIOGRAF was unable to produce a result.

Figure 4 is a plot that compares the energy calculation

times for BIOGRAF, using no cutoffs, with the time to do the same calculation on the MasPar. As expected, for small atoms, the SGI is much faster: it has a 50MHz processor and 32-bit internal data path, whereas the MasPar MP-1 has a 12.5MHz processor and 4-bit data path in the PEs. The plot (which has a log scale on the vertical axis) also shows the expected linear growth of the execution time of the MasPar program as a function of the number of atoms. For this set of data the BIOGRAF execution is worse than expected; it should grow quadratically but instead is steeper. Most of this time is accounted for by the hydrogen bond interaction term; when this term is removed, the BIOGRAF plot is closer to quadratic and it does not exceed the MasPar time until the number of atoms is greater than 1500.

7 Summary and Future Work

This paper describes the design and implementation of a data-parallel algorithm for computing the potential energy in a biological molecule using the Dreiding force field. The time to compute the energy in a molecule with n atoms is $O(n)$ on n processors. The parallel algorithm allows one to compute all nonbonded interactions, which would require time $O(n^2)$ on a single processor. Ignoring nonbonded terms for atoms that are beyond a small (10Å) cutoff distance can lead to errors as large as 15%, so it would appear to be worthwhile to compute all interactions. The parallel algorithm, when implemented on a MasPar MP-1, begins to be more effective (faster and more accurate) than a sequential algorithm running on an SGI Indigo workstation on molecules with about 750 atoms.

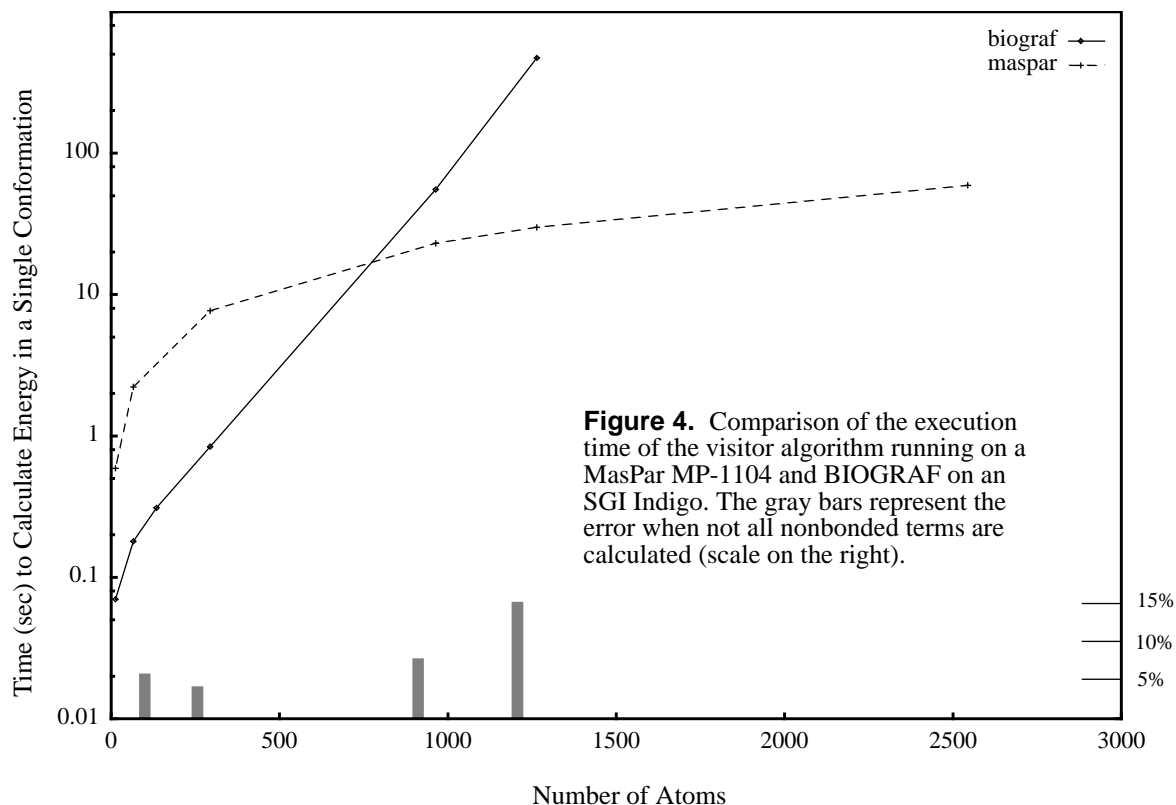


Figure 4. Comparison of the execution time of the visitor algorithm running on a MasPar MP-1104 and BIOGRAF on an SGI Indigo. The gray bars represent the error when not all nonbonded terms are calculated (scale on the right).

Our future plans are to optimize the algorithm so it can be used as part of an energy minimization procedure. In a minimization program the structure of the molecule (its bonds, angles, and torsions) needs to be computed only once; after that a single pass around the virtual ring will suffice to calculate the energy of each new conformation. A straightforward method for using the visitor algorithm on the MasPar will be to run a standard sequential optimization code on the DECstation front-end and have it invoke the parallel algorithm to compute the energy in each new conformation. A longer range project will be to investigate a parallel minimization technique.

The visitor algorithm should be easy to implement on a general parallel machine. It will be simple to code it for SPMD execution in a language such as Dataparallel C [3] or High Performance Fortran and let compilers for those languages allocate atoms to processors. With a little extra work, for example figuring out how to “pipeline” a batch of atoms from one processor to the next while processors are working on a different set of atoms, it should be possible to implement the method in a message passing style and get it to run efficiently on a distributed memory parallel processor or heterogeneous network of workstations.

8 References

- [1] BIOGRAF ver. 3.0, Molecular Simulations, Inc., Waltham, MA.
- [2] Dai, Z., Dauchez, M., Thomas, G., and Peticolas, W.L. *J. Biomolecular Structure and Dynamics* 9(6), pp. 1155–1183, 1992.
- [3] Hatcher, P. J. and Quinn, M. J. *Dataparallel Programming*. MIT Press, 1991.
- [4] Leighton, F. T. *Introduction Parallel Algorithms and Architectures: Arrays, Trees, and Hypercubes*. Morgan Kaufman Publishers, San Mateo, 1992.
- [5] Long, D.A. *Raman Spectroscopy*. McGraw-Hill, 1977.
- [6] Mayo, S.L., Olafson, B.D., and Goddard, W.A. III. *J. Phys. Chem.* 94, pp. 8897–8909, 1990.
- [7] Peticolas, W.L. and Evertsz, E. *Methods in Enzymology*, vol. 211, pp. 335–352, 1992.
- [8] Seitz, C. L. The Cosmic Cube. *Communications of the ACM* 28(1), Jan. 1985.