New Architectures for a New Biology

David E. Shaw

D. E. Shaw Research, LLC and Center for Computational Biology and Bioinformatics Columbia University
Background
(A Bit of Basic Biochemistry)
DNA Codes for Proteins

The 20 Amino Acids
Polypeptide Chain

Source: www.yourgenome.org
Levels of Protein Structure

Source: Robert Melamede, U. Colorado
What We Know and What We Don’t

- Decoded the genome
- Don’t know most protein structures
  - Especially membrane proteins
- No detailed picture of what most proteins do
- Don’t know how everything fits together into a working system
We Now Have The Parts List ...

<table>
<thead>
<tr>
<th>SECTION 7  PARTS LIST</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>7.1 BOARD A-I</strong></td>
</tr>
<tr>
<td>REFERENCE</td>
</tr>
<tr>
<td>A1, 2</td>
</tr>
<tr>
<td>A3, 4</td>
</tr>
<tr>
<td>Q6,8</td>
</tr>
<tr>
<td>Q2,3</td>
</tr>
<tr>
<td>Q9</td>
</tr>
<tr>
<td>Q1,4,5</td>
</tr>
<tr>
<td>CR1-6</td>
</tr>
<tr>
<td>P2</td>
</tr>
<tr>
<td>P1</td>
</tr>
<tr>
<td>T3, T4</td>
</tr>
<tr>
<td>T1</td>
</tr>
<tr>
<td>T2</td>
</tr>
<tr>
<td>C8</td>
</tr>
<tr>
<td>C2</td>
</tr>
<tr>
<td>C4, 5</td>
</tr>
<tr>
<td>S1</td>
</tr>
<tr>
<td>S2</td>
</tr>
</tbody>
</table>

| **7.2 BOARD B-I**     |
| REFERENCE | ARP PART NUMBER | ARP/MFG NUMBER |
| A1, 2     | 5601801        | A2801-008A     |
| A3        | 5601501        | A4024-006-2B   |
| Z1, 2     | 5602001        | A2803-002A     |
| Q1, 9, 10, 16, 17, 18 | 1301701 | 2N5172       |
| Q7, 14    | 1302801        | 2N6076         |
| Q4, 5, Q11/12 | 7502600 | APL027-008    |
| Q2, 8, 15 | 1302501        | 2N5461         |

| **7.3 BOARD C-I**     |
| REFERENCE | ARP PART NUMBER | ARP/MFG NUMBER |
| Q3        | 5600201        | A2803-003-1B   |
| Q6, 13    | 5600202        | A2803-003-2B   |
| CR1-3, 5-12 | 1200301       | 1N4148         |
| CR4       | 1200102        | 1N34           |
| C12, 16   | 1101201        | DM-15-681K     |
| C10, 11   | 1100612        | Tep-00-10/35-50-20 |
| R32, 44   | 1000105        | SA-21          |
| P16       | 5700701        | B2801-006-1D   |
| P5, 6, 7, 10, 11, 13, 14, 15, P1, 2, 3, 4, 8, 9, 12 | 5700702 | B2801-006-2B |
| T1, 4, 7  | 1000909        | U201R103B      |
| T2, 3, 5, 6 | 1000915      | U201R104B      |
| S1-11     | 1902401        | 01-481-0006    |

| REFERENCE | ARP PART NUMBER | ARP/MFG NUMBER |
| M1        | 4023           |                |
| A1        | 5601901        | A-2801-009-1   |
| A2        | 5601501        | B4023-006-2B   |
| Q12       | 1304601        | TZ81           |
| Q2, 3, 4, 6, 8, 10, 13, 16, 18, Q2, 5, 7, 9, 11, 14, 15, 17, 1301701 | 2N5172 | 2N6076 |
| CR1-22    | 1200301        | 1N4148         |
| C7, 8     | 1100602        | TAG-00-3.3/20-10/10 |
But We Don’t Know What the Parts Look Like ...
Or How They Fit Together ...
Or How The Whole Machine Works
How Can We Get There?

Two major approaches:

- **Experiments**
  - Wet lab
  - Hard, since everything is so small

- **Simulation**
  - Simulate:
    - How proteins fold (structure, dynamics)
    - How proteins interact with
      - Other proteins
      - Nucleic acids
      - Drug molecules
  - Gold standard: Molecular dynamics (MD)
Molecular Dynamics
Molecular Dynamics

Divide time into discrete time steps

$t$ →

~1 fs time step
Molecular Dynamics

Calculate forces

Molecular mechanics
force field
Molecular Dynamics

Move atoms
Molecular Dynamics

Move atoms

... a little bit
Molecular Dynamics

Iterate

... and iterate

... and iterate

Integrate Newton's laws of motion
Example of an MD Simulation
Main Problem With MD

Too slow!

Example I just showed:

- 2 ns simulated time
- 3.4 CPU-days to simulate
Goals and Strategy
Thought Experiment

- What if MD were
  - Perfectly accurate?
  - Infinitely fast?

- Would be easy to perform
  arbitrary computational experiments
  - Determine structures by watching them form
  - Figure out what happens by watching it happen
  - Transform measurement into data mining
Two Distinct Problems

*Problem 1:* Simulate many short trajectories

*Problem 2:* Simulate one long trajectory
Simulating Many Short Trajectories

- Can answer surprising number of interesting questions

- Can be done using
  - Many slow computers
  - Distributed processing approach
  - Little inter-processor communication

- E.g., Pande’s *Folding at Home* project
Simulating One Long Trajectory

- Harder problem
- Essential to elucidate many biologically interesting processes
- Requires a single machine with
  - Extremely high performance
  - Truly massive parallelism
  - Lots of inter-processor communication
Our Goal

- Single, millisecond-scale MD simulations
  - Protein with 64K atoms
  - Explicit water molecules

- Why?
  - That’s the time scale at which many biologically interesting things start to happen
Protein Folding

Image: Istvan Kolossvary & Annabel Todd, D. E. Shaw Research
Interactions Between Proteins

Binding of Drugs to their Molecular Targets

Mechanisms of Intracellular Machines

What Will It Take to Simulate a Millisecond?

- We need an enormous increase in speed
  - Current (single processor): ~ 100 ms / fs
  - Goal will require < 10 \( \mu s / fs \)

- Required speedup:
  > 10,000x faster than current single-processor speed
  ~ 1,000x faster than current parallel implementations
Target Simulation Speed

3.4 days today
(one processor)

~ 13 seconds on our machine
(one segment)
### Molecular Mechanics Force Field

\[
E = \sum_{\text{bonds}} k_b (r - r_0)^2 \quad \text{Stretch}
\]

\[
+ \sum_{\text{angles}} k_{\theta} (\theta - \theta_0)^2 \quad \text{Bend}
\]

\[
+ \sum_{\text{torsions}} A [1 + \cos(n\tau - \phi)] \quad \text{Torsion}
\]

\[
+ \sum_{i} \sum_{j>i} \frac{q_i q_j}{r_{ij}} \quad \text{Electrostatic}
\]

\[
+ \sum_{i} \sum_{j>i} \frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^6} \quad \text{Van der Waals}
\]
Stretch Term

\[ E = \sum_{\text{bonds}} k_b (r - r_0)^2 \]
The stretch term of potential energy is given by the equation:

\[ E = \sum_{\text{bonds}} k_b (r - r_o)^2 \]
Distance Between Centers of Atoms

Stretch Term

\[ E = \sum_{\text{bonds}} k_b (r - r_0)^2 \]
Stretch Term

\[ E = \sum_{\text{bonds}} k_b (r - r_0)^2 \]
Distance Between Centers of Atoms

Potential Energy

Stretch Term

\[ E = \sum_{\text{bonds}} k_b (r - r_0)^2 \]
Distance Between Centers of Atoms

Potential Energy

Stretch Term

\[ E = \sum_{\text{bonds}} k_b (r - r_0)^2 \]
Distance Between Centers of Atoms

Potential Energy

Stretch Term

\[ E = \sum_{\text{bonds}} k_b \left( r - r_0 \right)^2 \]
**Bend Term**

$$E = \sum_{\text{angles}} k_\theta (\theta - \theta_0)^2$$

![Diagram](diagram.png)
Bend Term

\[ E = \sum_{\text{angles}} k_\theta (\theta - \theta_0)^2 \]
Bend Term

\[ E = \sum_{\text{angles}} k_\theta (\theta - \theta_0)^2 \]
Bend Term

\[ E = \sum_{\text{angles}} k_\theta (\theta - \theta_0)^2 \]
Bend Term

$E = \sum_{\text{angles}} k_\theta (\theta - \theta_0)^2$
Bond Angle Potential Energy

\[ E = \sum_{\text{angles}} k_\theta (\theta - \theta_0)^2 \]
Bend Term

\[ \theta = \theta_0 \]

\[ E = \sum_{\text{angles}} k_\theta (\theta - \theta_0)^2 \]
Torsion Term

\[ E = \sum_{\text{torsions}} A [1 + \cos(n\tau - \varphi)] \]
Torsion Term

\[ E = \sum_{\text{tortions}} A[1 + \cos(n\tau - \phi)] \]
Torsion Term

\[ E = \sum_{\text{torsions}} A[1 + \cos(n\tau - \varphi)] \]
Electrostatic Term

\[ E = \sum_i \sum_{j>i} \frac{q_i q_j}{r_{ij}} \]
Electrostatic Term

\[ E = \sum_i \sum_{j>i} \frac{q_i q_j}{r_{ij}} \]

Distance Between Centers of Atoms

Potential Energy
Electrostatic Term

\[ E = \sum \sum \frac{q_i q_j}{r_{ij}} \]

Potential Energy vs. Distance Between Centers of Atoms
Electrostatic Term

\[ E = \sum_{i} \sum_{j>i} \frac{q_i q_j}{r_{ij}} \]
Electrostatic Term

\[ E = \sum_{i} \sum_{j > i} \frac{q_i q_j}{r_{ij}} \]

Distance Between Centers of Atoms
Electrostatic Term

\[ E = \sum_{i} \sum_{j \neq i} \frac{q_i q_j}{r_{ij}} \]
Van der Waals Terms

Potential Energy

Distance Between Centers of Atoms

\[ E = \sum_i \sum_{j>i} \frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^6} \]
Van der Waals Terms

Potential Energy

Distance Between Centers of Atoms

\[ E = \sum_i \sum_{j>i} \left( \frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^6} \right) \]
Van der Waals Terms

Potential Energy versus Distance Between Centers of Atoms

The potential energy $E$ is given by:

$$E = \sum_{i} \sum_{j>i} \left( \frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}} \right)$$

- **Repulsive** ($1/r^{12}$)
- **Attractive** ($1/r^6$)
- **Combined**
Van der Waals Terms

\[ E = \sum_i \sum_{j>i} \frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^6} \]
Van der Waals Terms

\[ E = \sum_i \sum_{j>i} \frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^6} \]

- **Attractive** \((1/r^6)\)
- **Repulsive** \((1/r^{12})\)
- **Combined**

Potential Energy

Distance Between Centers of Atoms
Van der Waals Terms

Potential Energy

Distance Between Centers of Atoms

\[ E = \sum_i \sum_{j>i} \frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^6} \]

- **Repulsive** \((1/r^{12})\)
- **Attractive** \((1/r^6)\)
- **Combined**
Van der Waals Terms

\[ E = \sum_i \sum_{j>i} \frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^6} \]

- **Repulsive** \( \frac{1}{r^{12}} \)
- **Attractive** \( \frac{1}{r^6} \)
- **Combined**
Molecular Mechanics Force Field

\[ E = \sum_{\text{bonds}} k_b (r - r_0)^2 \]  
Stretch

\[ + \sum_{\text{angles}} k_\theta (\theta - \theta_0)^2 \]  
Bend

\[ + \sum_{\text{torsions}} A[1 + \cos(n\tau - \varphi)] \]  
Torsion

\[ + \sum_i \sum_{j>i} \frac{q_i q_j}{r_{ij}} \]  
Electrostatic

\[ + \sum_i \sum_{j>i} \frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^6} \]  
Van der Waals
What Takes So Long?

- Inner loop of force field evaluation looks at all pairs of atoms (within distance $R$)
- On the order of 64K atoms in typical system
- Repeat $\sim 10^{12}$ times
- Current approaches too slow by several orders of magnitude
- What can be done?
Our Strategy

- **New architectures**
  - Designing a specialized machine
  - Enormously parallel architecture
  - Based on special-purpose ASICs
  - Dramatically faster for MD, but less flexible
  - Projected completion: 2008

- **New algorithms**
  - Applicable to
    - Conventional clusters
    - Our own machine
  - Scale to very large # of processing elements
Interdisciplinary Lab

Computational Chemists and Biologists

Computer Scientists and Applied Mathematicians

Computer Architects and Engineers
Alternative Machine Architectures

- Conventional cluster of commodity processors
- General-purpose scientific supercomputer
- Special-purpose molecular dynamics machine
Conventional Cluster of Commodity Processors

- **Strengths:**
  - Flexibility
  - Mass market economies of scale

- **Limitations**
  - Doesn’t exploit special features of the problem
  - Communication bottlenecks
    - Between processor and memory
    - Among processors
  - Insufficient arithmetic power
Typical Commodity Microprocessor
Typical Commodity Microprocessor
General-Purpose Scientific Supercomputer

- E.g., IBM *Blue Gene*

- More demanding goal than ours
  - General-purpose scientific supercomputing
  - Fast for wide range of applications

- Strengths:
  - Flexibility
  - Ease of programmability

- Limitations for MD simulations
  - Expensive
  - Still not fast enough for our purposes
Our Special-Purpose MD Machine

- Strengths:
  - Several orders of magnitude faster for MD
  - Excellent cost/performance characteristics

- Limitations:
  - Not designed for other scientific applications
    - They’d be difficult to program
    - Still wouldn’t be especially fast
  - Limited flexibility
Source of Speedup on Our Machine

- Judicious use of **arithmetic specialization**
  - Flexibility, programmability only where needed
  - Elsewhere, hardware tailored for speed
    - Tables and parameters, but not programmable

- Carefully **choreographed communication**
  - Data flows to just where it’s needed
  - Almost never need to access off-chip memory
Two Subsystems on Each ASIC

- Programmable, general-purpose
- Efficient geometric operations

Flexible Subsystem

Specialized Subsystem

- Pairwise point interactions
- Enormously parallel
Where We Use Specialized Hardware

Specialized hardware (with tables, parameters) where:

- Inner loop
- Simple, regular algorithmic structure
- Unlikely to change

Examples:
- Electrostatic forces
- Van der Waals interactions (at least attractive term)
Example: Particle Interaction Pipeline (one of 32)
Array of 32 Particle Interaction Pipelines

Step 1

Step 2

Step 3

Step 4

Step 5

Step 6
Advantages of Particle Interaction Pipelines

- Save area that would have been allocated to
  - Cache
  - Control logic
  - Wires

- Achieve extremely high arithmetic density

- Save time that would have been spent on
  - Cache misses,
  - Load/store instructions
  - Misc. data shuffling
Where We Use Flexible Hardware

- Use programmable hardware where:
  - Algorithm less regular
  - Smaller % of total time
    - E.g., local interactions (fewer of them)
  - More likely to change

- Examples:
  - Bonded interactions
  - Bond length constraints
  - Experimentation with
    - New, short-range force field terms
    - Alternative integration techniques
Forms of Parallelism in Flexible Subsystem

- The Flexible Subsystem exploits three forms of parallelism:
  - Multi-core parallelism
  - Instruction-level parallelism
  - SIMD parallelism
Overview of the Flexible Subsystem

Tensilica Core (w/ Custom Instructions)

Tensilica Core (w/ Custom Instructions)

Tensilica Core (w/ Custom Instructions)

Tensilica Core (w/ Custom Instructions)

GC = Geometry Core (each a VLIW processor)
Geometry Core
(one of 8; 64 pipelined lanes/chip)
System-Level Organization

- Multiple segments (probably 8 in first machine)
- 512 nodes (each with one ASIC) per segment
  - Organized in an 8 x 8 x 8 toroidal mesh
- Topology reflects physical space being simulated:
  - Three-dimensional nearest neighbor connections
  - Periodic boundary conditions
3D Torus Network
But Communication is Still a Bottleneck

- Scalability limited by inter-chip communication

- To execute a *single* millisecond-scale simulation,
  - Need a huge number of processing elements
  - Must dramatically reduce amount of data transferred between these processing elements

- Can’t do this without fundamentally new algorithms
The NT Algorithm
Range-Limited Pairwise Particle Interactions

- Efficient methods known for distant interactions
- Pairwise, non-bonded interactions dominate
- Range-limited $n$-body problem
New Algorithm

- Parallel algorithm for range-limited $n$-body problem
- Called the NT (for “Neutral Territory”) Method *
- Asymptotically less inter-processor communication than traditional spatial decomposition methods
- Constant factors also very attractive
  - Significant improvements on typical cluster
  - Major win on large machines

* Shaw, J. Comp. Chem. 26, Oct. 2005
Desirable Properties

- Ideally, a parallel algorithm for the range-limited $n$-body problem would:
  - Exploit the range limitation to reduce computational load
  - Scale such that data transfer approaches zero as $p \to \infty$
Asymptotic Comparison With Traditional Spatial Decomposition Methods

- NT Method has both of these properties:

<table>
<thead>
<tr>
<th></th>
<th>Exploitable range limitation</th>
<th>Scaling with number of processors</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Traditional methods</strong></td>
<td>$O(R^3)$ neighbors</td>
<td>Not scalable</td>
</tr>
<tr>
<td><strong>NT Method</strong></td>
<td>$O(R^{3/2})$ neighbors</td>
<td>$O(P^{-1/2})$ scaling</td>
</tr>
</tbody>
</table>
Partitioning of Space Into Boxes

Atom $A$

Home box of atom $A$
Two-Dimensional Analog of the NT Method

Traditional Method (2D Analog)

NT Method (2D Analog)

Green = interaction box; blue = import region
How can it be better to meet on neutral territory?

Traditional Method (2D)  NT Method (2D)

Number of pairwise interactions (~ product of areas)

Number of atoms imported (~ sum of areas):
Actual 3D Algorithm

- Considerably more complex
  - Odd number of dimensions introduces complications

- Can be made to work
  - Math gets more complicated
  - Performance advantage just as large

- Start by describing 3D version of traditional spatial decomposition methods
Traditional 3D Spatial Decomposition Methods
Traditional Spatial Decomposition Method
Interaction Box and Import Region

Green = Interaction box  Blue = Import region
Site of Interaction, Traditional Method

- Interact
  - One atom from (cubical) interaction box
  - One atom from either interaction box or import region

- All interactions occur within home box of one of the two atoms

- How much inter-processor communication?
Import Subregion Face ($-\chi$)
Import Subregion
Edge (−x, +z)
Import Subregion Corner \((+x, -y, +z)\)
Import Volume, Traditional Method

- Import region of traditional spatial decomposition method:
  - 3 face subregions
  - 6 edge subregions
  - 4 corner subregions

\[ 3Rb^2 + 3\pi R^2 b/2 + 2\pi R^3/3 \]

where \( b = \) side length of (cubical) box

- In limit as \( p \in \mathcal{O} \), import volume approaches

\[ 2\pi R^3/3 \]
The Three-Dimensional NT Algorithm
NT Method
Interaction Box and Import Region

Green = Interaction box  Blue = Import region
The Tower
(outer tower in blue)
The Plate
(outer plate in blue)
Site of Interaction, NT Method

- Interact
  - One atom from tower
  - One atom from plate

- Both atoms may have to be imported

- They meet “on neutral territory”
Aspect Ratio Optimization in NT

- Dimensions of box ⇒ dimensions of tower, plate

- Volume of box determined by
  - Size of molecular system
  - Number of processors

- Aspect ratio of box is free parameter
  - $x$ and $y$ dimensions equal; ratio to $z$ can vary
  - Optimize to minimize communication

- Optimal aspect ratio depends on number of processors
  - More processors ⇒ shorter, fatter box (balance)
Scaling of the NT Method
64 Processors

Assumes 50,000 atoms, interaction radius = 12\text{A}, density = 0.1 \text{atom/A}^3
Scaling of the NT Method
512 Processors

Assumes 50,000 atoms, interaction radius = 12A, density = 0.1 atom/A³
Scaling of the NT Method
4K Processors

Assumes 50,000 atoms, interaction radius = 12Å, density = 0.1 atom/Å³
Scaling of the NT Method
32K Processors

Assumes 50,000 atoms, interaction radius = 12Å, density = 0.1 atom/A³
NT’s Import Volume With Cubical Box

- Import volume:

\[ V_i = 2Rb_{xy}^2 + 2Rb_{xy}b_z + \frac{\pi R^2 b_z}{2} \]

where \( b_{xy} = x \& y \) dimensions of box
\( b_z = z \) dimension of box

- Optimize ratio \( \frac{b_{xy}}{b_z} \) to minimize import volume

4 face subregions
2 edge subregions
*No* corner subregions
NT: Optimal Aspect Ratio and Import Volume

Results:

- Optimal $b_{xy} = \left[ c^{1/2} + (V_b c^{-1/2} - c)^{1/2} \right] / 2$

  where $c = d/6 - 2\pi RV_b/d$

  $d = \left\{ 27 V_b^2 - 3 \left[ 3 V_b^3 \left( (4\pi R)^3 + 27 V_b \right) \right]^{1/2} \right\}^{1/3}$

  $V_b =$ box volume

- To find minimal import volume:
  - Use optimal $b_{xy}$ to calculate optimal $b_z$
  - Substitute into equation for $V_i$
NT’s Import Volume With Optimized Box

- Limit as $p \to \infty$:
  \[ V_i = 2\pi^{1/2} R^{3/2} V_b^{1/2} \]

- $V_b \sim N/p$, where $N$ is # atoms in molecular system

- So $V_i = O(R^{3/2} (N/p)^{1/2})$
Comparison of Traditional and NT Methods
Traditional Method Imports Corner Subregions
NT Method Doesn’t Import Any Corner Subregions
NT vs. Traditional Method

- Traditional spatial decomposition method:
  - Transfer time $\sim$ volume of sphere of radius $R$ (for large $p$)

- NT method
  - Transfer time $\sim$ square root of that sphere’s volume

- Advantage of NT over traditional method grows as number of processors increases
Scaling of Traditional vs. NT Method
64 Processors

Assumes 50,000 atoms, interaction radius = 12Å, density = 0.1 atom/Å³
Scaling of Traditional vs. NT Method
512 Processors

Assumes 50,000 atoms, interaction radius = 12A, density = 0.1 atom/A³
Scaling of Traditional vs. NT Method
4K Processors

Assumes 50,000 atoms, interaction radius = 12Å, density = 0.1 atom/Å³
Scaling of Traditional vs. NT Method
32K Processors

Assumes 50,000 atoms, interaction radius = 12Å, density = 0.1 atom/Å³
Assumes 50,000 atoms, interaction radius = 12Å, density = 0.1 atom/Å³
Time unit is time required to import data associated with one atom
An Open Question That Keeps Me Awake at Night
Are Force Fields Accurate Enough?

- Nobody knows how accurate the force fields that everyone uses actually are
  - Can’t simulate for long enough to know
  - If problems surface, we may at least be able to
    - Figure out why
    - Take steps to fix them

- But we already know that fast, single MD simulations will prove sufficient to answer at least some major scientific questions