Simulative Analysis of a Multidimensional Torus-based Reconfigurable Cluster for Molecular Dynamics

Abhijeet Lawande, Hanchao Yang, Alan D. George, Herman Lam
NSF Center for High-Performance Reconfigurable Computing (CHREC)
Department of Electrical and Computer Engineering
University of Florida
Gainesville, FL 32611-6200
lawande@chrec.org, mannesy@ufl.edu, george@chrec.org, hlam@chrec.org

Abstract—**Molecular dynamics (MD) is a large-scale, communication-intensive problem that has been the subject of high-performance computing research and acceleration for years. Not surprisingly, the most success in accelerating MD comes from specialized systems such as the Anton machine. Our goal is to design a reconfigurable system that can accelerate MD while also being amenable to other communication-intensive applications. In this paper, we present a performance model for the 3D FFT kernel that forms the core of MD simulation on Anton. We validate the model against published Anton performance data and use the data to design and evaluate a similar interconnect for our existing Novo-G reconfigurable supercomputer. Through simulation studies, we predict that the upgraded machine will achieve nearly double the performance of Anton and fifty times that of established clusters like BlueGene/L for the 3D FFT kernel.**

Keywords-High-performance computing; Computer simulation; Reconfigurable architectures; Field-programmable gate arrays

I. Introduction

The field of computational science deals with the development of mathematical models to depict natural processes and solve scientific problems. Unlike laboratory experimentation, computational science is limited by the models available, and the resources required to run them. It has been successfully applied to fields such as bioinformatics, fluid dynamics, and quantum mechanics [1] to build and test computational models of a complexity that cannot easily be replicated in the laboratory. A common theme in the above areas is the N-body problem: simulation of the interaction between multiple bodies due to the forces between them. At the molecular level, this simulation is termed molecular dynamics (MD) and can be applied to areas of science as varied as biomolecular engineering, neurobiology, material science, and nanotechnology. While originally restricted to use on general-purpose systems, a number of specialized systems have since been developed for MD, such as MDGRAPE [2] and Anton [3], [4].

Anton was developed in 2008 as a specialized system designed to accelerate MD simulation by several orders of magnitude and bring millisecond-scale simulations of tens of thousands of atoms within reach [5]. While Anton has achieved this goal, the system is only available in limited quantity, can only be used for MD simulations, and incurs a high non-recurring engineering cost. Our goal is to design a system that can perform as well as Anton on problems like MD, while retaining the ability to accelerate other large-scale computational applications as well. To that end, the use of reconfigurable-computing technology such as field-programmable gate arrays (FPGAs) is central to our theme of performance and flexibility.

FPGAs have been widely used to accelerate scientific-computing applications. The ability to reconfigure an FPGA's internal fabric to suit the application's needs means that many applications that perform poorly on traditional architectures are amenable for acceleration on an FPGA. Surveying the top 10 supercomputers in the world today [6], we see that 4 of 10 use accelerators in the form of Intel Xeon Phi coprocessors or Nvidia GPUs. The longer learning curve and turnaround time of FPGA designs have limited their adoption in this respect. Even so, at our center we have had much documented success with our FPGA-centric cluster, Novo-G [7], consisting of nearly 400 Stratix III and Stratix IV FPGAs from Altera. Leveraging our success in application acceleration in bioinformatics [8]–[10], image processing [11]–[13] and financial [14] domains among others, we plan to upgrade Novo-G to target communication-intensive applications that would benefit from FPGA acceleration.

The Anton machine was designed with a low-latency, 3D torus interconnect to help accelerate MD and its components. The component that interests us the most is the 3D FFT kernel that accounts for a large section of the MD computation and communication time. To understand its behavior better, we developed a simulation model of 3D FFT execution on Anton. VisualSim (www.mirabilisdesign.com/new/visualsim), a commercial discrete-event simulation and modeling tool, was used as it provides a simple graphical interface and a variety of basic blocks with which to construct models. The model was validated against published Anton performance data with less than 7% error and used to design and evaluate a similar interconnect for Novo-G via simulation. While the model is designed with MD in mind, many applications would benefit from a low-latency, direct interconnect between FPGAs. This reconfigurable cluster, which we have dubbed Novo-G# (novo-gee-sharp), is predicted to have better 3D FFT performance than Anton while using smaller system sizes, and can outperform conventional systems like the BlueGene/L by an order of magnitude. The performance improvement is attributed not just to the interconnect, but also the higher computational density on our devices, and optimizations to the 3D FFT communication pattern on Novo-G#.

To summarize, the objective of this work is the acceleration of molecular dynamics simulation on specialized hardware, with special attention to the 3D FFT kernel optimized for the Anton machine [15]. The methodology used is to model execution of the 3D FFT on Anton, and leverage the developed model to design an efficient inter-FPGA interconnect for Novo-G. The model is also used to predict FFT execution time for the resulting Novo-G# system.

The remainder of this paper is organized as follows. Section II gives background information on molecular dynamics, the Anton machine, and Novo-G. Section III describes our approach to modeling 3D FFT running on the Anton architecture. Section IV introduces the design for Novo-G# and its modeling in VisualSim, along with predicted 3D FFT performance. Finally, Section V concludes the paper.

II. Background

In this section, we briefly review the literature on the molecular-dynamics problem along with an overview of the Anton machine and its architecture. We also describe the development of the Novo-G reconfigurable supercomputer.

A. *Molecular dynamics and the FFT kernel*

Molecular dynamics is a computer simulation that models the physical behavior at the molecular or atomic level. MD simulation is frequently used in scientific fields such as biomolecular engineering, neurobiology, and material science. It is well-known for its computation intensity and demanding design requirements for accuracy. Generally, several CPU days to CPU months are needed for a dynamic simulation of DNA or protein molecules, ranging from nanoseconds to microseconds [16]. Thus, an accuracy-sufficient and time-efficient MD simulation is of importance to research in those fields.

MD software packages, such as NAMD [17] and GROMACS [18], are available for various platforms and scales. In general, the MD algorithm is composed of two parts: computing the interactive forces among particles in the simulation; and deriving their positions and velocities through the integration of those forces. Generally, the performance bottleneck is the force calculation step [19]. The total force for each simulated particle is computed as the sum of bonding forces, which depends on covalent bond structure of particles, and non-bonding forces, which involves the electrostatic and Van der Waals interactions between all pairs of particles in the system.

Calculating the non-bonded forces requires a pairwise computation for every pair of particles in the system. Since the fall-off rate for the Van der Waals force is considerably greater than that of the electrostatic force, a cutoff radius can be applied, thereby simplifying the computation. For long-range forces, such as the electrostatic force, other approximation methods need to be used to reduce the computation time. Among them, the Particle Mesh Ewald (PME) [20] and k-space Gaussian Split Ewald (k-GSE) [21] are the most popular. Both methods use a volumetric (3D) FFT to simplify the computation of electrostatic forces. Benchmark results from [19] using GROMACS on various system sizes show that the scalability of long-range force calculation is worse than that of range-limited force calculation, thus making long-range force calculation the most time-intensive part of MD in high-parallelism cases.

A 3D FFT calculation of size $N_x \times N_y \times N_z$ is composed of multiple 1D FFTs and can be divided into three stages: computing $N_y \times N_z$ 1D FFTs of size N_x in the X dimension; then $N_x \times N_z$ 1D FFTs of size N_y in the Y dimension; and finally $N_x \times N_y$ 1D FFTs of size N_z in the Z dimension. The influence of the 3D FFT calculation is negligible when compared to the number of messages required for this parallel implementation, thus limiting the communication scalability of long-range force calculation.

B. *The Anton machine*

Many efforts have been made to accelerate MD simulation on multi-node systems (e.g., BlueGene/L [22] and MDGRAPE [2]). Among them, Anton, a special-purpose parallel supercomputer stands out because of its low-latency communication network. Each Anton node is an application-specific integrated circuit (ASIC) designed specifically for MD. Unlike the MDGRAPE systems that divide MD computation between the host and ASIC processors, on Anton all MD computation takes place on the ASICs. Anton has also been shown to outperform modern high-performance computing systems due to fast, low-latency, inter-node communication. The data in Table I, collected from various references, compares the performance of Anton with other platforms. Here, a node is the basic element of the system, consisting of a single processing unit (ASIC or CPU) and associated components. The efficient use of hardware in Anton speeds up Dihydrofolate Reductase (DHFR) simulation by several orders of magnitude.

Table I. Comparison of MD performance[a] on various platforms

MD System	Ref.	Nodes	Real time per step (μs)
Anton	[4]	512	19
Desmond on 2.4 GHz AMD	[19]	256	1,400
Blue Matter on BG/L	[22]	8,192	1,700
NAMD on 2.4 GHz AMD	[17]	128	6,300
MDGRAPE-3	[2]	12	26,000
GROMACS on 2.4 GHz AMD	[3]	1	181,000

[a] Measured for Dihydrofolate Reductase (DHFR) simulation. Table adapted from [3].

Fig. 1 describes the basic node architecture of the Anton machine. A typical system consists of 512 nodes, with all the nodes connected in an 8×8×8 torus network. Each node is connected to its six neighbors in six directions with 50.6 Gbps bidirectional links. Each node uses a High-Throughput Interaction Subsystem (HTIS) to calculate range-limited interactions, perform charge spreading, and perform force interpolation. A flexible subsystem is responsible for long-range force calculation, particle updates, and the remainder of the MD pipeline. Each node also includes accumulation memories to sum forces and charges, and a 256-bit bidirectional intra-node ring network that facilitates data movement.

In a typical MD simulation, Anton spends almost 60% of its time in computing long-range forces [17]. Acceleration of the 3D FFT kernel is therefore the focus of this paper. Each flexible subsystem in an Anton node contains four processing slices and eight Geometry cores (GCs) that compute the individual 1D FFT stages of the 3D FFT kernel.

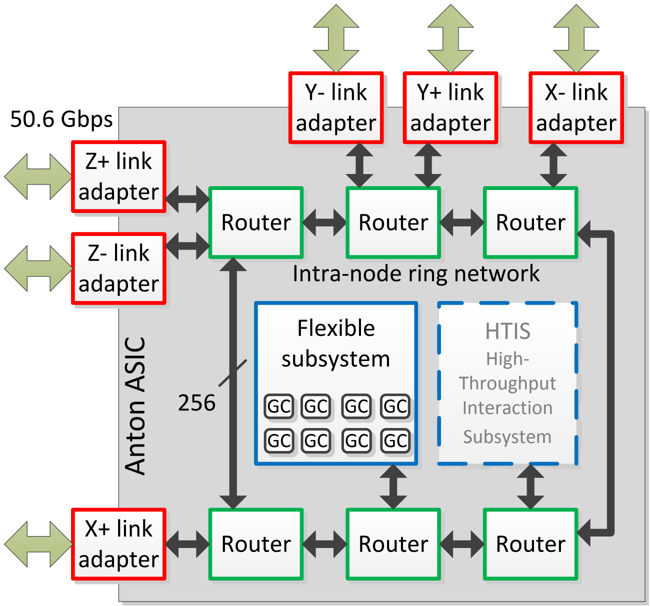


Fig. 1. Architecture of Anton machine's ASICs depicting the intra-node ring network. GC: Geometry Core responsible for 1D FFT computation.

C. Development of Novo-G

Novo-G [7] began in 2009 as an effort to create a research cluster using high-density FPGA boards to accelerate scientific applications. The machine began with a head node and 24 Linux servers, each featuring a quad-FPGA board from GiDEL (www.gidel.com/products), for a total of 96 Altera Stratix III E260 FPGAs. Over subsequent years, the machine has been upgraded annually and now stands at 192 Stratix III FPGAs in 24 servers and another 192 Stratix IV E530s housed in 12 servers. Each server features dual Intel Xeon multicore processors that connect to the FPGA boards via PCI Express. Connectivity is provided by gigabit Ethernet and DDR InfiniBand within the system, and a 10 Gb/s connection to the Florida LambdaRail.

Novo-G has been the platform of choice for a variety of application-acceleration projects undertaken by the NSF Center for High-Performance Reconfigurable Computing (CHREC). The following are some of the applications developed for Novo-G. Blast-Wrapped Smith-Waterman (bioinformatics) on 128 FPGAs shows a speedup of 50,000 against SSEARCH [8]. Image segmentation (image processing) on four FPGAs shows a speedup of 1,106 against an optimized serial baseline [11]. Monte Carlo options pricing (financial computing) on 48 FPGAs shows a speedup of 7,134 against an optimized serial baseline [14]. The one common factor among the above applications is that they are embarrassingly parallel and can therefore scale almost linearly with the available hardware resources. A greater challenge is that of accelerating communication-intensive applications like MD. Traditionally, this communication makes use of centralized networks such as Ethernet or InfiniBand and entails many interactions between the FPGA and the host. Our proposed system would feature a multidimensional network that connects the FPGAs in a 3D torus with bidirectional links, 6 links per FPGA, at 40 Gbaud per link. For comparison, 4X QDR InfiniBand also provides a signaling rate of 40 Gbaud. A large part of Anton's success with MD comes from the use of a low-latency 3D torus network between processors, and we intend to emulate this facet in Novo-G#.

III. Modeling The Anton Architecture

In order to develop a model for 3D FFT execution on Anton, we collect published data on the Anton architecture, operation, and decomposition of the 3D FFT algorithm. This section describes the process of modeling the 3D FFT algorithm and the Anton architecture in VisualSim. We validate the developed model against published run times of the 3D FFT on Anton for various system and 3D FFT sizes.

A. 3D FFT application modeling in VisualSim

As discussed earlier, computation of long-range forces in the MD application is more efficient in Fourier space than real space. The computation in the kernel consists of the Fourier transform of the charge-density function, which is multiplied by the transform of the potential function, followed by an inverse Fourier transform (IFFT) of the product. In keeping with published Anton data [15], we only model the FFT and IFFT stages.

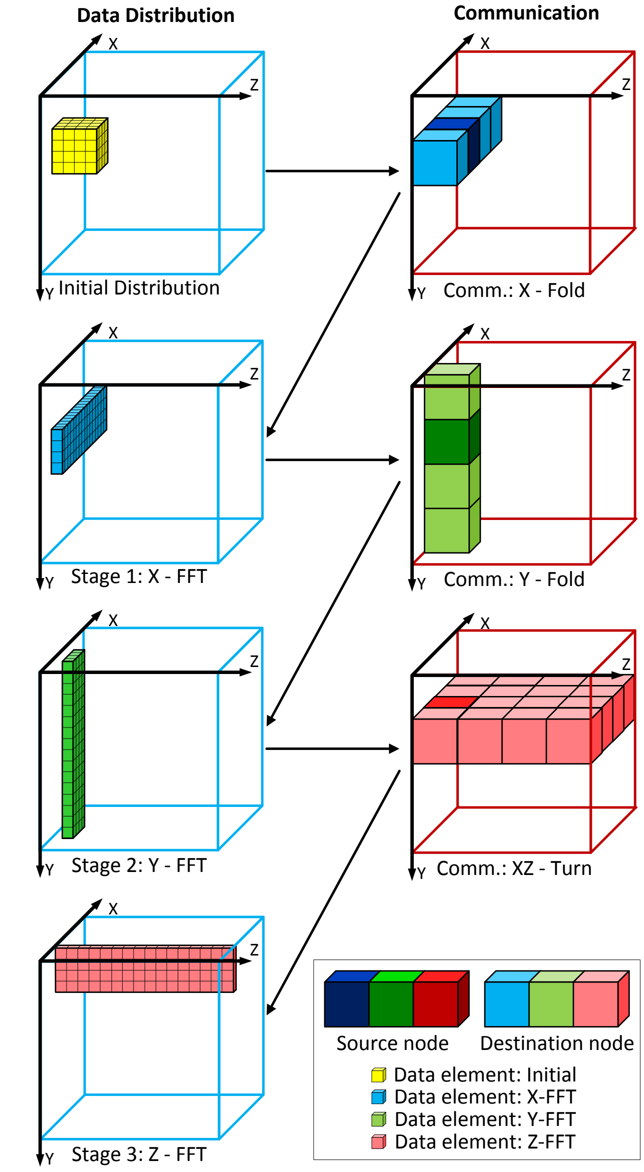


Fig. 2. Data cube for a 16×16×16 point FFT running on a 4×4×4 system. Data distribution is shown for a single node at Cartesian coordinates (1,1,0).

The communication behavior of the 3D FFT largely depends on how the data is distributed for each stage of the FFT. For a 16×16×16 data set processed on a 4×4×4 system, Fig. 2 depicts the communication pattern and FFT stages, using the node at Cartesian coordinates (1,1,0) as an example. To succinctly describe the communication pattern for a given FFT and system size, we use a modified form of the notation described in [15]. In this notation, a data point $(x_n \dots x_0, y_n \dots y_0, z_n \dots z_0)$, representing its 3D binary address, is mapped to the node with Cartesian coordinates $(Node\_x, Node\_y, Node\_z)$. If $index$ represents the execution unit within the node, and $offset$ represents the data offset within the execution unit, the final mapping for each data point is given by:

$$(x_n \dots x_0, y_n \dots y_0, z_n \dots z_0) \leftrightarrow (\text{Node\_x, Node\_y, Node\_z}).\text{index.offset}$$

As shown in Fig. 3a, this notation can be used to determine the data that needs to be transmitted between each node before an FFT computation can begin. The data pattern, and by extension the communication pattern, will change based on the size of the FFT and the system. For the specific case of a 32×32×32 FFT distributed on an 8×8×8 Anton system, two 1D FFTs are computed on each node in every stage.

$(x_4x_3x_2,y_4y_3y_2,z_4z_3z_2).x_1.x_0y_1y_0z_1z_0$	Original Distribution
$(y_1y_0z_0,y_4y_3y_2,z_4z_3z_2).z_1.x_4x_3x_2x_1x_0$	After X - Fold
$(x_1x_0z_0,x_4x_3x_2,z_4z_3z_2).z_1.y_4y_3y_2y_1y_0$	After XY - Turn
$(x_1x_0y_0,x_4x_3x_2,y_4y_3y_2).y_1.z_4z_3z_2z_1z_0$	After XZ - Turn

(a)

```
INIT: Distribute initial FFT data tokens to all nodes
BEGIN:
If input_token corresponds to current FFT stage
    Wait until all data elements arrive at node
    If Verify_mode then
        Compute local FFT
    End If
    For all Data elements in the node
        Generate message for next stage
        Determine destination and data index
        Transmit message delayed by FFT computation time
    End for
End If
```

(b)

Fig. 3. (a) Representation of data movement for 32×32×32 FFT running on 8×8×8 Anton; (b) Pseudocode for FFT application model in VisualSim.

In the VisualSim model, behavior of the 3D FFT is implemented as a script in a virtual machine block, the pseudocode for which is shown in Fig. 3b. Execution of the block is triggered each time an $input\_token$ is received. We use the $Verify\_mode$ flag to turn on code in the script that computes the FFT output. We can thus verify the model by computing the actual FFT outputs for random input data and comparing with the Matlab 3D FFT function. Once verified, the $Verify\_mode$ switch is turned off to speed up the model.

B. *VisualSim modeling of Anton*

Parameters that we use to model the Anton machine are collected from multiple publications, notably [3]–[5], [15], [23], [24], with preference given to recent publications. Table II summarizes these parameters. We built the model in VisualSim as a hierarchical model, with the node and channel models described as independent classes. This method allows us the flexibility of developing the node and channel models independently and modularly. At the top-level, an instance of each class is created for every node in the system being modeled.

TABLE II. MODELING PARAMETERS FOR ANTON MACHINE

Parameter		**Value**	**References**
System frequency		485 MHz	[5], [15], [24]
Internal bandwidth		124.2 Gbit/s	[4], [5], [23], [24]
External bandwidth		50.6 Gbit/s	[3]–[5], [23]
Synchronization delay		42 ns	[24]
Package writing delay		36 ns	[24]
Wire delay	x	4 ns	[24]
	y	8 ns	[24]
	z	10 ns	[24]
Transceiver delay		20 ns	[24]
FFT calculation time	1GCs	137 cycles	[15]
	4 GCs	75 cycles	[15]

Fig. 4 shows the final Anton model. The FFT algorithm and routing algorithm are implemented as scripts running inside virtual-machine blocks, while the remainder of the system is modeled using basic blocks from the VisualSim library. To reduce the complexity of the model, we converted the intra-chip ring network, a 256-bit bidirectional bus, to a delay model. To model contention among messages, separate queues are used for each direction of the ring, and for the ±Y and ±Z link pairs. Table III summarizes the delay for each source-destination pair.

TABLE III. ROUTING LATENCIES (ns) FOR ANTON MACHINE

Destination	**Source direction**						
	X+	***X-***	***Y+***	***Y-***	***Z+***	***Z-***	***Processing slice***
X+	—	31	25	25	19	19	19
X-	31	—	19	19	25	25	25
Y+	25	19	—	13	25	25	31
Y-	25	19	13	—	19	19	31
Z+	19	25	25	19	—	13	25
Z-	19	25	25	19	13	—	25
Processing slice	19	25	31	31	25	25	—

Anton uses Geometry Cores (GCs) to handle the basic FFT operation. Computing a 32-point 1D FFT on a single GC requires 137 clock cycles. Almost twice the performance (75 cycles) can be achieved by parallelizing every 1D FFT over four GCs using redundant computation [15]. Since 1D FFT is an $O(N \log N)$ function, we compute the execution time (assuming operation on 1 GC) for a 1D FFT of size $NumPoints$ as:

$$\text{Cycles}_{\text{NumPoints}} = 137 \times \frac{\text{NumPoints} \times \log_2 \text{NumPoints}}{32 \times \log_2 32}$$

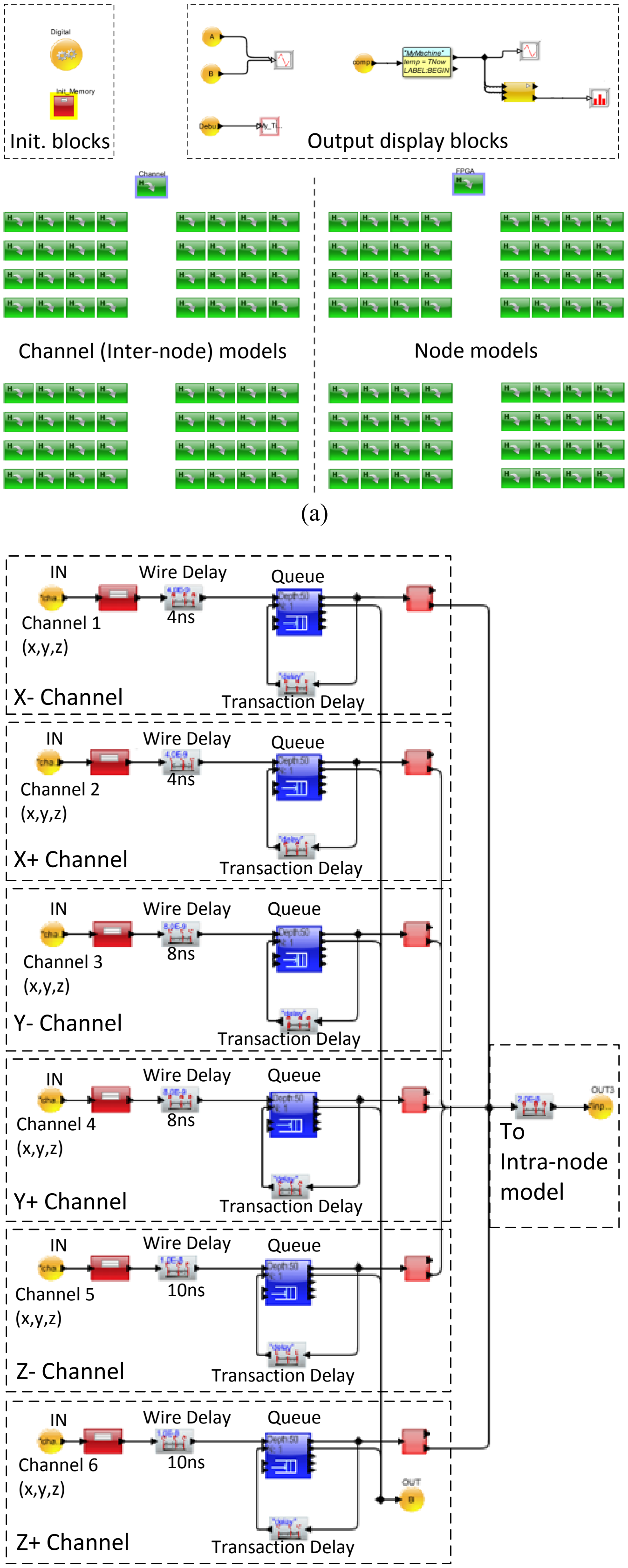
C. Model validation via Anton

We validate the Anton model by comparing simulation times from our model to those reported in [15] by researchers using the Anton machine. The referenced paper provides 3D FFT execution times averaged over 10 runs of a 3D FFT followed by a 3D IFFT. We follow the same procedure in our VisualSim runs.

TABLE IV. MODEL VALIDATION WITH ANTON DATA

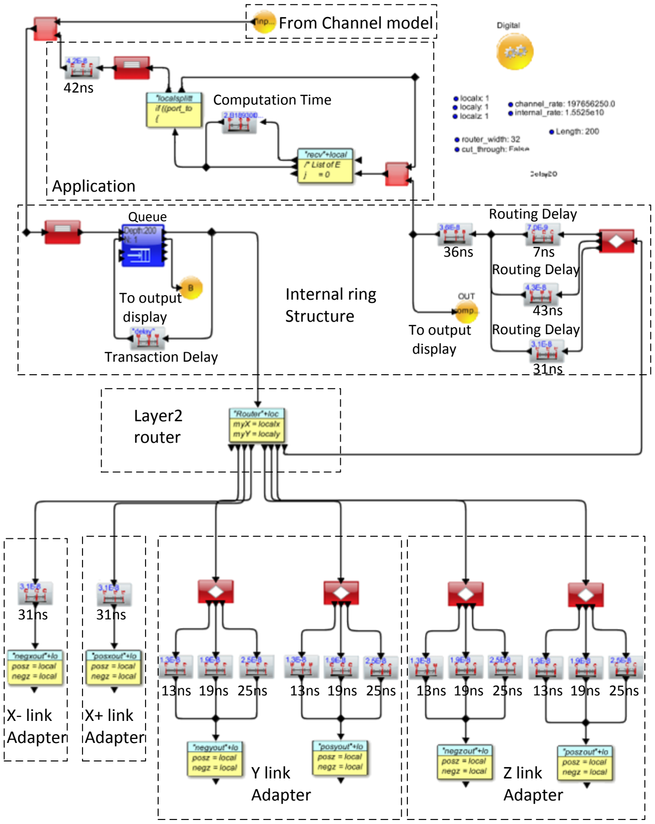
System size	FFT size	Parallel strategy	Anton exec. time (µs)	Measured model time (µs)	Error
8×8×8	32×32×32	1FFT:4GCs	3.7	3.5	5.4%
	32×32×32	1FFT:1GC	4.0	3.75	6.25%
	64×64×64	1FFT:1GC	13.2	12.7	3.7%
4×4×4	16×16×16	1FFT:1GC	2.4	2.55	6.25%
	32×32×32	2FFTs:1GC	10.5	10.0	4.8%

Simulation runs of the Anton model are summarized in Table IV. Since we are leveraging published Anton data, the number of data points available is limited. However, we believe that the FFT size, system size, and parallelization strategy shown here are sufficiently diverse to expose any errors in the model. Given that the FFT data distribution changes with the problem and system size, the communication patterns are different in each case. Special attention is paid to the 64×64×64 FFT, where the communication pattern is optimized (as described in [15]) to reduce the load on the communication network. In all cases, we observe that the prediction error of our model is less than 7%.



(a)

(b)



(c)

Fig. 4. VisualSim model of Anton machine: (a) top-level model for 4×4×4 system size; (b) Channel (Inter-node) model; (c) ASIC (Intra-node) model.

IV. The Novo-G# System

In this section, we describe the planned upgrades to the existing Novo-G infrastructure to add a 3D torus network that provides direct communication between FPGAs. The low-latency, high-bandwidth network is inspired by the model built in Section III and is shown to greatly improve the performance of communication-intensive applications. We also describe benchmarking and simulation tests of the Novo-G# system components, followed by development of the VisualSim model. The high degree of similarity between the Novo-G# model and the Anton model validated previously allows us to estimate Novo-G# system performance using the Anton model as a reference.

A. *Development of Novo-G#*

The system targeted here is part of our effort to create an FPGA cluster that can handle communication-intensive applications like MD. The on-going upgrade features Stratix V GX FPGAs from Altera, which are optimized for high-bandwidth applications and support up to 36 on-chip transceivers with operation up to 14.1 Gbaud. GiDEL has provided invaluable assistance by designing accelerator boards and a custom daughterboard that allows access to 24 such transceivers. The transceivers are grouped into six bidirectional links, each link consisting of four parallel channels, enabling the construction of a 3D torus of arbitrary size. The initial deployment of the prototype system will have 32 nodes housed in eight chassis and form a 2×4×4 torus. Each node also houses two multicore CPUs that can communicate with the boards over an 8-lane PCIe 3.0 interface.

B. *3D FFT implementation for Stratix V devices*

The parameters required to model the prototype system were derived from several sources: benchmarking of two prototype GiDEL Stratix V boards in our lab; RTL simulation of the 3D FFT core and protocol stack; and system specifications and datasheets. Table V summarizes these parameters. We designed a basic protocol stack based on the block diagram in Fig. 5, with FFT cores instantiated through Altera MegaWizard [25], and simulated the system using Modelsim from Mentor Graphics.

A switch (layer 2) and a router (layer 3) based on dimension-order routing were implemented to provide the appropriate addressing and routing services required by the application block. Flow and congestion control between FPGAs was provided by using a backpressure channel between the input and output queues on each node. While better distributed-routing strategies exist, the objective of the model is to mimic the latency between parts of the node. The model also provides a framework for exploration of better network designs, which is critical to realizing the predicted system performance on the physical system.

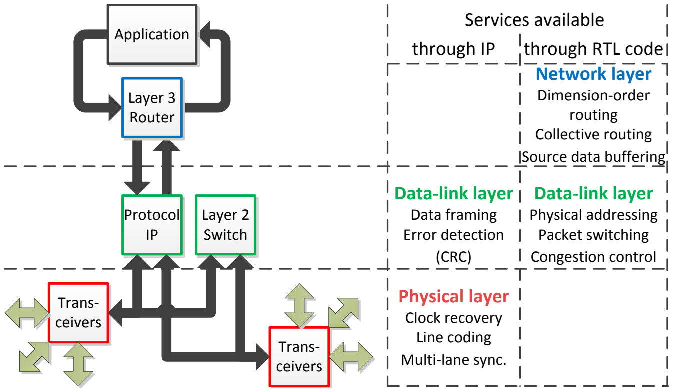


Fig. 6. Block diagram of protocol stack for prototype system. Services provided to the user through RTL or third-party IP are also shown.

Table V. Modeling parameters for prototype system

	Parameter	Value	Notes
From H/W benchmarks	System frequency	250 MHz	Derived from Altera MegaCore data for Stratix V devices; N = FFT Size
	FFT latency	N cycles	
	Num_cores	16	
	Propogation delay	20 ns	From roundtrip latency
	Channel rate	10 Gbps	Data rate per channel of a link
	Channel width	4	No. of physical channels per link
From H/W simulation	Internal latency for completed packets	11 cycles	Altera FIFOs in critical data path contribute 3 cycles each (optimized for frequency)
	Internal latency for incomplete packets	11 cycles	
	Write packet initiation delay	9 cycles	
	Internal bidirectional bus width	256 bits	User-selectable parameter

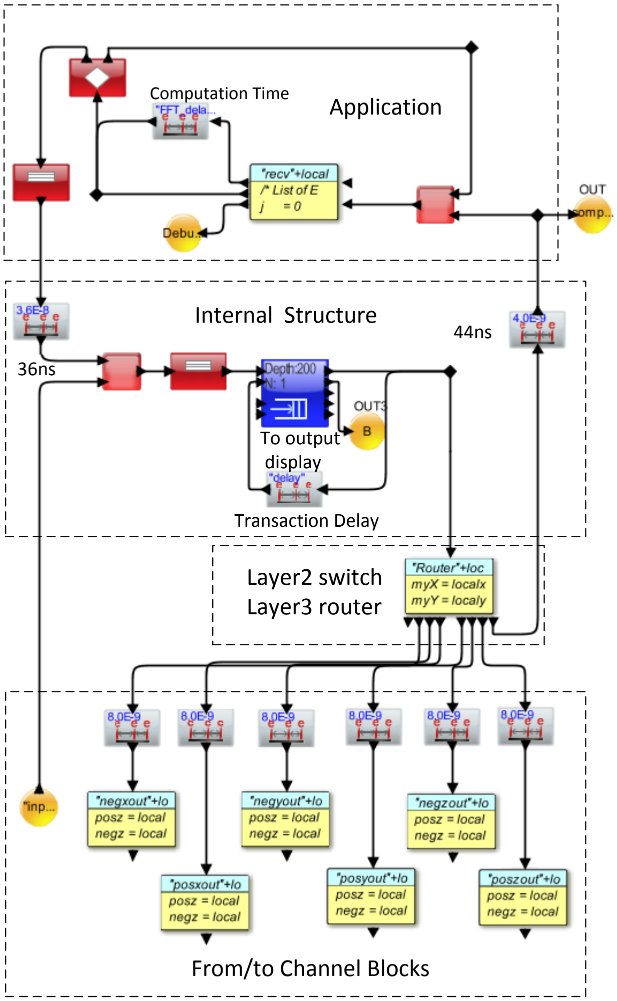


Fig. 5. Updated node model for prototype system. The top-level and channel models structurally remain the same, they are not shown here.

TABLE VI. PREDICTED 3D FFT KERNEL EXECUTION TIMES (μs)

FFT size	*System size*				
	2×2×2	*2×4×2*	*2×4×4*	*4×4×4*	*4×8×4*
16×16×16	1.753	1.574	1.509	1.570	1.678
32×32×32	9.997	8.563	5.749	3.943	3.302
64×64×64	75.94	64.47	42.31	26.11	17.71
128×128×128	603.5	511.8	334.8	207.2	136.7

The physical parameters for the prototype system were obtained by benchmarking a pair of Stratix V boards interconnected using a proprietary cable. At the physical layer (layer 1), the Interlaken protocol was used to provide clock recovery, 64b/67b encoding for DC balance, and synchronization of the four serial channels per link. At the data-link layer (layer 2), the Altera Serialite III protocol was used to provide framing and error handling. The Serialite III protocol combined with the RTL code used in the Modelsim simulation forms the basic protocol stack that provides network services to the application.

C. *VisualSim modeling of prototype system*

In order to leverage our developed prototype system model for prediction, structural changes between the Novo-G# model and the Anton model are kept to a minimum. The primary change visible in Fig. 6 is the absence of the intra-chip ring network. In this case, the script in the Virtual Machine block also handles intra-node routing. The Novo-G# parameters from Table V are also applied to the model.

The target system size under consideration ranges from 8 to 128 nodes. A system size considerably smaller than Anton allows us to optimize the FFT data distribution further. In most of the FFT stages, data movement between pairs of nodes can be consolidated as described in [15]. This optimization results in a coarse-grained communication pattern that is more suitable for Novo-G#, while also reducing the incurred packet and routing overheads. In the case of the 32×32×32 FFT, now distributed on a 4×4×4 system, the communication pattern can also be modified to use a y-fold (Fig. 7) instead of an xy-corner-turn (Fig. 3b), thus reducing the communication load on the system. Conversely, the smaller system size also leads to longer calculation times since the distribution of 1D FFTs per node is higher. The final model retains the modularity of the original and can easily be modified to match future optimizations to the prototype system. It can also be used to evaluate design changes before they are implemented on the actual system. We plan to use the model to evaluate better intra-node routing architectures and inter-FPGA communication protocols.

$(x_4x_3,y_4y_3,z_4z_3).x_2x_1y_2z_2.x_0y_1y_0z_1z_0$	Original Distribution
$(z_2z_1,y_4y_3,z_4z_3).y_2y_1y_0z_0.x_4x_3x_2x_1x_0$	After X - Fold
$(z_2z_1,x_4x_3,z_4z_3).x_2x_1x_0z_0.y_4y_3y_2y_1y_0$	After Y - Fold
$(x_2x_1,x_4x_3,y_4y_3).x_0y_2y_1y_0.z_4z_3z_2z_1z_0$	After XZ - Turn

Fig. 7. Representation of optimized data movement for 32×32×32 FFT running on 4×4×4 system.

D. *Model prediction for prototype system*

We can now use the Novo-G# model to predict FFT/IFFT kernel execution time on the prototype system. Table VI shows the total FFT/IFFT execution time over various FFT and system sizes. In each case, a communication pattern appropriate for the given FFT and system size is chosen.

TABLE VII. 3D FFT KERNEL EXECUTION ON DIFFERENT SYSTEMS

System	**No. of nodes**	**FFT size**	**Time (μs)**	**Novo-G# (pred.)**		**Est. speedup**
				nodes	***time (μs)***	
Anton	512	32x32x32	7.4	64	3.9	1.90
		64×64×64	26.4	128	17.7	1.49
BlueGene/L	1024	64×64×64	1000	128	17.7	56.5
		128×128×128	5000	128	136.7	36.6

Table VII compares predicted Novo-G# run time to Anton and BlueGene/L [22], [26] for larger FFT sizes. We chose BlueGene/L over newer systems because of the availability of published runtime data on the 3D FFT. Compared to Anton, we observe nearly twice the performance over a range of parameter values, at system sizes smaller than Anton. Predicted Novo-G# execution time is over fifty times better than BlueGene/L. The notable improvement in performance over Anton is attributed to a few factors. Firstly, the Stratix V devices use 28-nm fabrication, allowing much better computation density than Anton, and better clock frequencies than previous reconfigurable systems. Based on the Altera MegaCore resource utilization, we have limited each node to 16 FFT cores, which consume about 50% of the DSPs on each FPGA. Secondly, the inherently smaller system allows more data reuse between FFT stages. The tradeoff is that smaller systems require multiple FFT rounds per stage, increasing total run time. Finally, we use optimizations to reduce the total number of messages, which gives an advantage to small and medium systems.

Note that while scaling to larger Novo-G# sizes may be appealing for the reduced execution time, Fig. 8 shows that system utilization reduces dramatically with scale. Thus, there is a diminishing return on performance as more hardware resources are used. Finally, our model scales to larger problem and system sizes, but the memory allocation in VisualSim is insufficient to simulate larger models at this time.

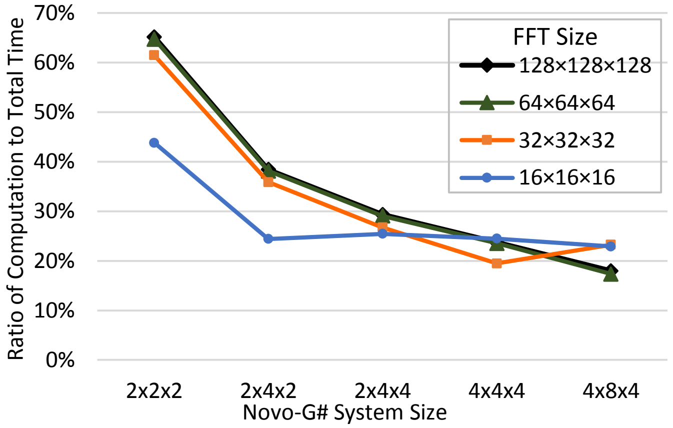


Fig. 8. Ratio of FFT computation time to total time for 3D FFT predicted from Novo-G# model for various FFT and system sizes.

V. Conclusions

The 3D or volumetric FFT is the dominant kernel of the molecular-dynamics application. MD being a frequently used, large-scale, data-movement problem, there have been many attempts to accelerate it on conventional clusters such as Blue-Gene/L and on unconventional systems such as Anton.

In this paper, we presented a discrete-event simulation model for the Anton machine that was validated against published execution times for the Anton machine within 7% error. We leveraged the model to design a low-latency 3D torus network for our existing Novo-G supercomputer and modified the model to represent a new Novo-G# system, and predicted its performance. Simulation studies showed 3D FFT execution on Novo-G# to be nearly twice as fast as Anton. The performance is predicted to be up to 56 times as fast when compared to conventional high-performance computing systems like Blue Gene/L.

A major factor that contributes to this performance is the low-latency, high-bandwidth network for Novo-G# that is inspired by Anton, and the computational density provided by the Stratix V devices. Combined with tight integration with the network interfaces and a flexible design, these factors result in a system that can successfully be used to accelerate MD and other computational science applications. Moving forward, the developed models will be used to design and evaluate better intra-node routing and protocols for use in Novo-G# with the goal of developing a reconfigurable and sustainable cluster for the acceleration of communication-intensive applications.

Acknowledgment

This work was supported in part by the I/UCRC Program of the National Science Foundation under Grant Nos. EEC-0642422 and IIP-1161022. We gratefully acknowledge tools and licenses provided by Mirabilis Design.

References

[1] A. Bourlioux et al., Eds., *Modern Methods in Scientific Computing and Applications*. Springer Netherlands, 2002.

[2] T. Narumi et al., "Gordon Bell finalists II---A 55 TFLOPS simulation of amyloid-forming peptides from yeast prion Sup35 with the special-purpose computer system MDGRAPE-3," in *Proceedings of the 2006 ACM/IEEE conference on Supercomputing - SC '06*, 2006, p. 49.

[3] D. E. Shaw et al., "Anton, a special-purpose machine for molecular dynamics simulation," in *Proceedings of the 34th annual international symposium on Computer architecture - ISCA '07*, 2007, p. 1.

[4] D. E. Shaw et al., "Anton, a special-purpose machine for molecular dynamics simulation," *Commun. ACM*, vol. 51, no. 7, p. 91, Jul. 2008.

[5] D. E. Shaw et al., "Millisecond-scale molecular dynamics simulations on Anton," in *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis - SC '09*, 2009, no. c, p. 1.

[6] "November 2013 | Top 500 Supercomputer Sites." [Online]. Available: http://www.top500.org/lists/2013/11/.

[7] A. George et al., "Novo-G: At the Forefront of Scalable Reconfigurable Supercomputing," *Comput. Sci. Eng.*, vol. 13, no. 1, pp. 82–86, Jan. 2011.

[8] B. C. Lam et al., "BSW: FPGA-accelerated BLAST-Wrapped Smith-Waterman aligner," in *2013 International Conference on Reconfigurable Computing and FPGAs (ReConFig)*, 2013, pp. 1–7.

[9] M. Pascoe, C., Lawande, A., Lam, H., George, A., Sun, Y., Farmerie, W., & Herbordt, "Reconfigurable supercomputing with scalable systolic arrays and in-stream control for wavefront genomics processing," in *Proc. of Symposium on Application Accelerators in High-Performance Computing*, 2010, pp. 13–15.

[10] C. Pascoe et al., "FPGA-Accelerated Isotope Pattern Calculator for Use in Simulated Mass Spectrometry Peptide and Protein Chemistry," in *2012 Symposium on Application Accelerators in High Performance Computing*, 2012, pp. 111–120.

[11] S. Craciun et al., "A scalable RC architecture for mean-shift clustering," in *2013 IEEE 24th International Conference on Application-Specific Systems, Architectures and Processors*, 2013, pp. 370–374.

[12] A. Zicari, P and Lam, H and George, "Reconfigurable Computing Architecture for Accurate Disparity Map Calculation in Real-Time Stereo Vision," in *International Conference on Image Processing, Computer Vision, and Pattern Recognition*, 2013.

[13] S. Craciun et al., "A parallel hardware architecture for information-theoretic adaptive filtering," in *2010 FOURTH INTERNATIONAL WORKSHOP ON HIGH-PERFORMANCE RECONFIGURABLE COMPUTING TECHNOLOGY AND APPLICATIONS (HPRCTA)*, 2010, pp. 1–10.

[14] R. Sridharan et al., "FPGA-Based Reconfigurable Computing for Pricing Multi-asset Barrier Options," in *2012 Symposium on Application Accelerators in High Performance Computing*, 2012, pp. 34–43.

[15] C. Young et al., "A 32x32x32, spatially distributed 3D FFT in four microseconds on Anton," in *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis - SC '09*, 2009, no. c, p. 1.

[16] B. J. Alder and T. E. Wainwright, "Studies in Molecular Dynamics. I. General Method," *J. Chem. Phys.*, vol. 31, no. 2, p. 459, 1959.

[17] J. C. Phillips et al., "Scalable molecular dynamics with NAMD.," *J. Comput. Chem.*, vol. 26, no. 16, pp. 1781–802, Dec. 2005.

[18] H. J. C. Berendsen et al., "GROMACS: A message-passing parallel molecular dynamics implementation," *Comput. Phys. Commun.*, vol. 91, no. 1–3, pp. 43–56, Sep. 1995.

[19] K. Bowers et al., "Scalable Algorithms for Molecular Dynamics Simulations on Commodity Clusters," in *ACM/IEEE SC 2006 Conference (SC'06)*, 2006, pp. 43–43.

[20] T. Darden et al., "Particle mesh Ewald: An N·log(N) method for Ewald sums in large systems," *J. Chem. Phys.*, vol. 98, no. 12, p. 10089, 1993.

[21] Y. Shan et al., "Gaussian split Ewald: A fast Ewald mesh method for molecular simulation.," *J. Chem. Phys.*, vol. 122, no. 5, p. 54101, Feb. 2005.

[22] R. Fitch, BlakeG. and Rayshubskiy, Aleksandr and Eleftheriou, Maria and Ward, T.J.Christopher and Giampapa, Mark and Zhestkov, Yuri and Pitman, MichaelC. and Suits, Frank and Grossfield, Alan and Pitera, Jed and Swope, William and Zhou, Ruhong and Feller, Sc, "Blue Matter: Strong Scaling of Molecular Dynamics on Blue Gene/L," in *Computational Science – ICCS 2006*, Springer Berlin Heidelberg, 2006, pp. 846–854.

[23] J. S. Kuskin et al., "Incorporating flexibility in Anton, a specialized machine for molecular dynamics simulation," in *2008 IEEE 14th International Symposium on High Performance Computer Architecture*, 2008, pp. 343–354.

[24] R. O. Dror et al., "Exploiting 162-Nanosecond End-to-End Communication Latency on Anton," in *2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, 2010, pp. 1–12.

[25] "FFT Megacore Function | User Guide." [Online]. Available: http://www.altera.com/literature/ug/ug_fft.pdf.

[26] M. Eleftheriou and B. Fitch, "Performance measurements of the 3d FFT on the Blue Gene/L supercomputer," in *Euro-Par 2005 Parallel Processing: 11th International Euro-Par Conference*, 2005, no. 1, pp. 795–803.