LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-CONF-553873

Toward Real-time Modeling of Human Heart Ventricles at Cellular Resolution: Multi-hour Simulation of Drug-induced Arrhythmias

A. A. Mirin, D. F. Richards, J. N. Glosli, E. W. Draeger, B. Chan, J. Fattebert, W. D. Krauss, T. Oppelstrup, J. J. Rice, J. A. Gunnels, V. Gurev, C. Kim, J. Magerlein, M. Reumann, H. Wen

May 1, 2012

Supercomputing 2012
Salt Lake City, UT, United States
November 10, 2012 through November 16, 2012

Disclaimer

Toward Real-time Modeling of Human Heart Ventricles at Cellular Resolution: Simulation of Drug-induced Arrhythmias

Arthur A. Mirin, David F. Richards, James N. Glosli, Erik W. Draeger,
Bor Chan, Jean-luc Fattebert, William D. Krauss, Tomas Oppelstrup
Lawrence Livermore National Laboratory, Livermore, CA, USA
John Jeremy Rice, John A. Gunnels, Viatcheslav Gurev, Changhoan Kim,
John Magerlein, Matthias Reumann*, Hui-Fang Wen
IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA
*IBM Research Collaboratory for Life Sciences, Melbourne, AU

***Abstract*—We have developed a highly efficient and scalable cardiac electrophysiology simulation capability that supports groundbreaking resolution and detail to elucidate the mechanisms of sudden cardiac death from arrhythmia. We can simulate thousands of heartbeats at a resolution of 0.1 mm, comparable to the size of cardiac cells, thereby enabling scientific inquiry not previously possible. Based on scaling results from the partially deployed Sequoia IBM Blue Gene/Q machine at Lawrence Livermore National Laboratory and planned optimizations, we estimate that by SC12 we will simulate 8–10 heartbeats per minute — a time-to-solution 400–500 times faster than the state-of-the-art. Performance between 8 and 11 PFlop/s on the full 1,572,864 cores is anticipated, representing 40–55 percent of peak. The power of the model is demonstrated by illuminating the subtle arrhythmogenic mechanisms of anti-arrhythmic drugs that paradoxically increase arrhythmias in some patient populations.**

I. Introduction

Sudden cardiac death (SCD) from cardiac arrest is the most common cause of death worldwide, accounting for over 50 percent of all deaths from cardiovascular disease and 250,000 to 300,000 deaths annually in the United States [1]. While much controversy exists, SCD is generally accepted to result from arrhythmias, predominately ventricular fibrillation, and hence can be considered an electrical dysfunction of the heart. During a normal heartbeat, Purkinje fibers, a specialized impulse conduction system, trigger the heart ventricles to contract in a uniform and orderly fashion. Each triggering event produces activation (termed *depolarization*) of the ventricular heart cells, followed by a recovery (termed *repolarization*) with a delay of approximately 300–500 ms. In contrast, an arrhythmia is thought to arise from a reentrant activation that follows an abnormal circular path and produces a continuous and self-regenerating cycle of activation.

Despite more than 50 years of intense research, the exact underlying mechanisms of arrhythmogenesis remain elusive. Many factors are known to predispose one to SCD, but no one has been able to define precisely the sequence of events that precipitate the fatal arrhythmia that some have termed the *electrical accident* [2]. Current experimental techniques cannot resolve initiation events at the cellular level, and *in situ* measurements in an active beating heart will remain a challenge in the foreseeable future. For this reason, we believe that detailed cardiac simulations have the greatest potential to illuminate the details of the formation and persistence of reentrant activation patterns. Understanding gained from high fidelity simulations will help to better apply existing therapies and spur the development of new therapies with greater efficacy and fewer deleterious effects.

Detailed cardiac models include biophysically accurate representation of cells, the assembly of cells into anatomically accurate organ-level structures, and models of the cell-to-cell electrical coupling including preferred electrical conduction along the complex muscle fiber structure of the heart. Previous cardiac simulation implementations are limited in biophysical detail of the cellular models, size of the model structure (e.g., rabbit vs. human heart), spatial resolution, and practical length of simulation time. Published scientific studies typically cite spatial resolutions of 0.2–0.3 mm and 20–32 million active elements [3]–[6] and simulation times of at most tens of heartbeats.

To address these limitations, we have developed *Cardioid*, a high-performance cardiac simulation tool with detailed cell representations and support for heart geometries with near-cellular spatial resolution based on full-size human ventricle anatomy [7]. We use *Cardioid* to simulate the application of a class III anti-arrhythmic drug, which can unexpectedly increase arrhythmias in some patients [8]. This study, which includes application and clearance of the drug and the sporadic arrhythmia events that occur in certain circumstances [9], requires both 0.1 mm resolution and simulation of up to 5,000 heartbeats, a task beyond the capability of other codes.

To meet the extraordinary computational demands of such a study, *Cardioid* has been highly optimized for execution on the *Sequoia* IBM Blue Gene®/Q (BG/Q) machine at Lawrence Livermore National Laboratory (LLNL). We predict a factor of 400–500 throughput improvement over the state-of-the-art, based on results of [10], [4] and [11] projected to our

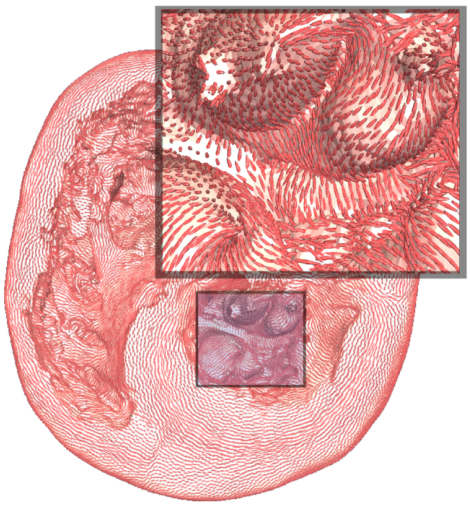

Fig. 1. Top view of ventricles looking down on the chambers. The inset shows details of the fine structures of the endocardium surface, including papillary muscles. The illustrated cardiac cells that are in red (not to scale) show the fiber direction.

planned simulations. Through use of low-level communication protocols, we attain near-perfect weak scaling. With judicious use of cache and registers, minimization of memory accesses, utilization of hardware threading, hardware assisted barriers for intranode synchronization, and exploitation of the SIMD instruction set, we are able to attain 43% of peak performance per computational node despite having only 1600 floating point operations per grid cell per iteration. This translates to over 4.3 Pflop/s on half the machine (already demonstrated) and over 8.6 Pflop/s projected on the full 1,572,864 core system (based on the demonstrated near-perfect weak scaling). This represents a speedup of a factor of 50 with respect to the unoptimized code. With further optimization we hope to exceed 10 Pflop/s on the full Sequoia.

The computational model is described in section II. In section III we describe the LLNL BG/Q system. The optimization strategy is discussed in section IV. Performance and scalability are discussed in section V. We discuss scientific capability and results in section VI. The conclusions are presented in section VII.

II. Computational Model

The electrophysiology of human heart ventricles is simulated as the ensemble behavior of individual cells that are physically and electrically connected (see Figure 1). Cells become electrically active because charged ionic species pass in and out of them through specialized protein structures called ionic channels. On every heartbeat, each cell activates and deactivates with a spike in the electrical potential of the cell membrane called an action potential (AP, see example in Figure 6). The AP's are transmitted from cell to cell via specialized gap junctions that pass ionic currents, and the cells are arranged into fibers with increased conduction in the axial direction.

To model this behavior there are two primary approaches. The first, more accurate method, is the bidomain model where the intra- and extra-cellular regions of the heart are modeled by coupled non-linear reaction-diffusion equations. The second method, called the monodomian model, is a simplification of the bidomain model using the assumption that the conductivities in the intra- and extra-cellular regions are proportional to each other. We employ the monodomain approximation which is very commonly used in the field because the results are sufficiently accurate without the additional cost of the bidomain method [12].

In the monodomain model the time evolution of the electrical potential of the cell membrane (V_m) is given by

$$C_m \frac{\partial V_m}{\partial t} = \frac{1}{\beta} \nabla \cdot (D \nabla (V_m)) - I_{\text{ion}} + I_{\text{stim}} \qquad (1)$$

with a zero-flux condition imposed at the heart tissue boundary. C_m is the membrane capacitance, β is the surface to volume ratio, and I_{stim} is the external stimulus current. The conductivity D is a spatially varying anisotropic tensor and is determined by the fiber structure of the heart. I_{ion} is the sum of all ion currents, and a variety of models for these currents have been proposed. Such models are called tissue models or reaction models and define the state of a heart cell in terms of values of ionic species concentrations and gate variables, where a gate expresses the probability that a particular ion channel is open. The evolution of the state variables (concentrations and gates) is expressed as a set of coupled nonlinear ODE's. The ionic currents for a given cell are independent of those of the other cells. The coupling between cells arises through the diffusion term of equation (1).

In 2006 ten Tusscher, et al. [13] developed a reaction model (hereafter referred to as TT06) in which the state of a cell is defined by 5 ion concentrations and 13 gates. The TT06 model simulates three types of cells: epicardial (epi), M and endocardial (endo) cells. Epi and endo cells are located respectively near the outside and inside walls of the heart. M cells, generally thought to be located in the mid-myocardium, are characterized by their unique repolarization properties. We have modified the TT06 model to add a late sodium current with an associated gate to better reproduce the observed variation of action potential duration (APD) in various cell types (see section VI). We refer to this enhanced model as TT06e.

The ODE's in TT06e can be divided into two classes. The time derivative for 12 of the 14 gate variables can be expressed simply as

$$\frac{dg_\beta}{dt} = (\mu_\beta(V_m) - g_\beta)\tau_\beta^{-1}(V_m) \qquad (2)$$

Notice that the evolution of each gate depends only on the value of the gate variable g_β and the membrane voltage, V_m. This observation is exploited in the optimization and discussed in section IV. The time derivative of the remaining degrees of freedom is expressed in the general form.

$$\frac{dg^*_\alpha}{dt} = F(\{g^*_\alpha\}, \{g_\beta\})$$

where F is a nonlinear function. For simplicity we refer to the first set of equations as gate equations and the remainder as non-gates.

We define a 3D Cartesian grid in which each grid cell contains either heart muscle tissue or does not. Our computational domain is defined as the union of all the tissue grid cells.

We discretize the diffusion term using a finite-volume approach, in which we replace the divergence operator with a surface integral over the six faces of each cell. Since the conductivity tensor D is a general anisotropic tensor, calculating the flux through each cell surface requires the evaluation of all three components of ∇V_m. The gradient is calculated using a simple finite difference algorithm (a 19-point stencil operator) on the values of V_m in the cells nearest to the surface considered. Setting the flux to zero through grid cell surfaces located at the boundary of the computational domain enforces the boundary conditions. If some of those neighboring cells are outside the computational domain, the finite difference schemes are modified to use only available values of V_m inside the computational domain, resulting in a stencil operator with weights dependent upon cell location. In practice, those weights are precomputed at the beginning of a run and stored for each cell. The time-integration is accomplished with a forward Euler scheme except for the so-called m-gate. For this gate τ can become quite small and an exponential integration suggested by Rush and Larsen [14] allows a much larger time step than forward Euler.

Our parallelization strategy consists of a hybrid approach where we make use of message passing and threading. We first partition the domain into a collection of subdomains, where each subdomain comprises a contiguous collection of cells. Subdomains are assigned to compute nodes in a one-to-one fashion. Due to the high overhead of MPI, our message passing is performed using lower-level primitives. The threading strategy is described in detail in section IV.

Load balance is accomplished at runtime using either of two approaches. The first approach is based on the method of Koradi, et al. [15], in which the domain is partitioned into a collection of Voronoi cells and balance achieved through minimizing a cost function. The second approach involves a weighted partitioning onto a three dimensional process grid, followed by a diffusive rebalancing between neighbors to equalize the number of cells per task. Each method has its strengths and is undergoing continued improvement; it is not yet clear if one is superior to the other.

III. Target Architecture

Sequoia, located at Lawrence Livermore National Laboratory, is the first and largest installation of the BG/Q architecture. When fully operational in June 2012, it will employ 98304 nodes and nearly 1.6 million processor cores to achieve a peak performance of roughly 20 PFlop/s. On a per-node basis, BG/Q will out-perform its predecessors Blue Gene/L and Blue Gene/P by a factor of 36 and 15, respectively.

Each BG/Q node is based on a system-on-a-chip processor fabricated with 45 nm technology [16]. By integrating processors, memory subsystem, and chip-to-chip communication subsystems on a single chip, the compute node achieves both outstanding power efficiency and reliability. Reliability is further enhanced by deploying state-of-the-art soft-error mitigation and error correction techniques throughout the design. Functionally, the chip contains 16 compute cores and 1 supplemental core to handle operating system tasks. Each core supports four-way simultaneous multithreading (SMT) and two-way concurrent instruction issue: one integer, branch, or load/store instruction and one floating-point instruction per clock cycle. Within a thread, dispatch, execution, and completion are in order. Four threads per core are sufficient to fully utilize the execution pipelines, prevent memory latency from adversely affecting performance, and avoid the necessity of out-of-order execution. Each core contains a quad SIMD double precision floating point unit (FPU) capable of 8 floating point operations (fused multiply add) per cycle. The 16 user processor cores at 1.6 GHz gives a BG/Q node a peak performance of 204.8 GFlop/s. Each processor core has a 16+16 kByte L1 instruction and data cache with a sophisticated pre-fetch. The processors share a central 32 MByte L2 cache with 563 GByte/s bisection bandwidth. The node main memory is 16 GByte of directly attached SDRAM-DDR3 providing 42.7GB/s total bandwidth.

The BG/Q L2 cache provides hardware support for transactional memory and speculative execution. These techniques facilitate execution of threaded code with potential data conflicts and dependencies, by allowing threads to run speculatively in parallel, while detecting and managing conflicts and dependencies as they occur. The L2 cache also supports atomic operations, which we use for thread synchronization. For operations such as locking and barriers, and for work-queue management, these atomic operations have much less latency than software implementations that use the traditional load-linked/store conditional or compare-and-swap mechanisms, especially for a large number of threads. Haring, et al. [16] gives an example of a 30x reduction in latency for 64 threads to obtain a ticket lock (where each thread obtains a unique number) using the atomic LoadIncrement instruction.

An integrated five-dimensional torus serves as the principal node-to-node communication network and also handles collectives and fast interrupts. Compared to the 3D torus architecture of its predecessor machines, the 5D torus in BG/Q markedly decreases the number of hops required to reach the farthest node in the machine from any given node, and thereby both increases the bandwidth and reduces the latency of node-to-node communication. The 5-D nearest neighbor exchange bandwidth is roughly 1.75 GByte/s per link with a latency of 700 ns. BG/Q is water cooled and has achieved a power efficiency of over 2 GFlop/s/Watt on a Linpack benchmark.

The Sequoia installation has one I/O node for every 128 compute nodes, theoretically supporting up to 4 Gbyte/s bandwidth per I/O node to a file system.

IV. Optimization strategy

In designing an optimization strategy for this problem, a number of significant challenges must be addressed. Chief among these is the relatively small amount of available work per core. Our discretization of a typical human heart with a grid resolution of 0.1 mm has about 370 million grid cells of heart tissue. On Sequoia this amounts to about 3800 cells per node, or about 250 per core. In this aggressive strong scaling regime, merely tuning C/C++ code to the BG/Q architecture is not enough. It is also necessary to modify the computational model to eliminate high-cost operations and expose concurrency while always maintaining fidelity to the underlying biology. Only this combination of hardware-aware and biology-aware tuning can achieve the best possible time-to-solution.

A. Biology-aware tuning

In TT06e there are 32 computationally expensive functions to be evaluated for each grid cell at each time step. These functions share the common form

$$f(V_m) = \frac{\sum_{i=0}^{N} A_i \exp(B_i V_m)}{\sum_{j=0}^{M} C_j \exp(D_j V_m)} \tag{3}$$

For example, μ and τ (see equation (2)) for the fGate are given by

$$\mu = \left(1 + \exp\left(\tfrac{V_m+20}{7}\right)\right)^{-1}$$

and

$$\frac{1}{\tau} = \frac{1.0 + \exp\left(\tfrac{25.0-V_m}{10.0}\right) + 80.0/\left(1.0 + \exp\left(\tfrac{V_m+30.0}{10.0}\right)\right)}{562.0 \exp\left(\tfrac{-(V_m+27.0)^2}{240.0}\right) + 31.0}$$

The majority of the computational effort in updating the TT06e state variables is associated with evaluation of these functions, so it is essential to find efficient ways to compute them.

The form and parameters in equation (3) are obtained not from any biological "first-principles", but are rather chosen to give reasonable limiting behavior and to fit experimental data which are subject to measurement error. Suppose form (3) is replaced with a lower-cost form such as a rational function approximate

$$f(V_m) \approx f_a(V_m) = \frac{\sum_{i=0}^{m} a_i V_m^i}{1 + \sum_{j=1}^{\ell} b_j V_m^j}$$

Such a replacement is justifiable as long as the limiting behavior is respected and the fit to data is equally good. We can in fact replace all 32 functions with rational function approximates with coefficients chosen to minimize the cost of evaluation while maintaining a relative error of $\epsilon = 10^{-4}$, where

$$\epsilon = \frac{\max_{V_m \in I} |f_a(V_m) - f(V_m)|}{\max_{V_m \in I} f(V_m) - \min_{V_m \in I} f(V_m)}, \; I = [-100, 50] \text{ mV}.$$

Approximates for each of the 1D functions are constructed separately, so the various rational function approximates contain polynomials of different order ranging from 1–18 in the numerator and 1–11 in the denominator.

One additional modification to the TT06e formulation is very useful. The rate of change of the potassium concentration K_i depends not only on the membrane potential, V_m, and other reaction state variables, S, but also on the rate of change of V_m due to the external stimulus, I_{stim} and diffusion, dV_m^D/dt:

$$\frac{dK_i}{dt} = -C[I_p(V_m(t), S(t)) + \frac{dV_m^D(t)}{dt} + I_{\text{stim}}(t)] \tag{4}$$

This introduces a coupling between the diffusion and reaction calculations. However by adding and subtracting the rate of change of the voltage due to reaction, dV_m^R/dt, equation (4) can be rewritten as;

$$\begin{aligned}\frac{dK_i}{dt} &= -C\left(Ip - \frac{dV_m^R(t)}{dt} + \frac{dV_m^R(t)}{dt} + \frac{dV_m^D(t)}{dt} + I_{\text{stim}}(t)\right) \\ &= -C\left(Ip - \frac{dV_m^R}{dt} + \frac{dV_m}{dt}\right)\end{aligned}$$

Rearranging terms and introducing the change of variable $V_m^K = K_i/C + V_m$ produces

$$\frac{dV_m^K}{dt} = -\left(Ip - \frac{dV_m^R}{dt}\right) = f(V_m(t), S(t))$$

and in terms of the new variable the coupling is removed.

B. Internode communication

With TT06e expressed in terms of rational function approximates, about 1600 floating point operations per grid cell are required to evaluate equation (1). Assuming 50% of peak performance and 3800 cells/node, this amounts to less than 70 μs of wall clock time. Because this is practically all of the computation in a time step, communication and synchronization operations must be optimized to be fast compared to 70 μs.

Internode communication is driven by the diffusion operator, which requires a halo exchange of ghost cells to satisfy the finite difference stencil. As we are well into the strong scaling regime, the surface to volume ratio is high (about 40–50%) and the message sizes are small. We implemented the halo exchange using MPI Isend and Irecv and found unacceptably high latencies of about 300 μs per time step. This forced us to abandon MPI calls and reimplement the halo exchange using the BG/Q low-level Systems Programming Interface (SPI).

Our optimized halo exchange is based on a remote-put strategy and takes advantage of the fact that communication descriptors whose specification includes message size, destination and routing can be constructed at initialization and associated with a first-in, first-out queue (FIFO) that is used for the remainder of the simulation. Communication is initiated by copying data from its regular storage location to the FIFO memory area and adjusting the tail pointer of the FIFO. This triggers the message engine to copy data to the network using the DMA instead of CPU resources. Message delivery completion is checked by monitoring counters for each message. On each node two receive buffers are used in alternation to avoid a race condition that could overwrite received data.

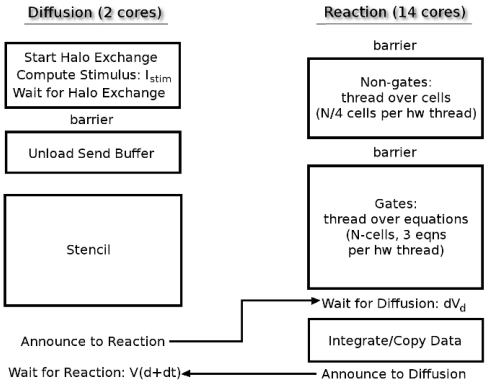

Fig. 2. Threading strategy. The diffusion and communication are shown on the left, and the reaction term on the right.

Large scale benchmarks on up to 48k nodes demonstrate that the SPI based halo exchange is completed in approximately 15 μs, a 20-fold improvement in communication time. Moreover, the CPUs are completely free during practically all of this time, allowing full opportunity to overlap computation. Other less frequent communication, such as initialization tasks and data aggregation for I/O, is performed with MPI.

C. Threading model

Initially we had planned to exploit the concurrency of multiple cores per node as well as multiple threads per core by decorating key loops with OpenMP pragmas. Unfortunately this also proved to be a source of unacceptable latency. On BG/Q the fork/join overhead for a single parallel for loop with 64 threads is nearly 15 μs. Hence, we are forced to abandon the model of using OMP pragmas to manage threads within our main time step loop. Instead, we begin a parallel section before the time step loop and manage the threads explicitly.

Our overall threading strategy is shown in Figure 2. The modifications to the potassium current described in Section IV-A allow us to run reaction and diffusion concurrently. Of the 16 cores on the node, we designate a small number of cores (presently 2) to be responsible for diffusion and the halo exchange. The remaining cores are responsible for the reaction model, time-integration loop, and most data copying/reformatting operations. The assignment of cores to diffusion or reaction is configurable at run-time. Synchronization between the diffusion and reaction threads and barriers within groups of threads are handled using the fast L2 atomic operations provided by the hardware, reducing associated overhead to under 1 μs.

There are several advantages to assigning threads in this manner. A principal consideration is that each core has its own L1 cache that is shared among the hardware threads. Assigning diffusion and reaction to separate cores simplifies cache management and allows us to balance the difference in computational intensity between diffusion and reaction by adjusting the number of cores assigned to the respective pieces. Additionally, SPI communication costs are covered by computation on 14 of 16 cores. Finally, since L2 bandwidth is on a per-core basis, copy operations are performed on the group with the larger number of cores.

D. Reaction calculation

The low level optimization of the reaction calculation is guided by the features of the BG/Q memory system and compute cores. Because of limited memory bandwidth (32 bytes/cycle/core from L1 and approximately 8 bytes/cycle/core from L2), data reuse is critical to prevent starvation of the quad SIMD FPU, which requires 96 input operand bytes/cycle for a quad fused multiply add (FMA). Such unfavorable bandwidth to Flop/s ratios are a typical feature of modern architectures. It is of course equally critical to "SIMDize" the calculation to exploit the vectorization potential of the SIMD instruction set. We chose the rational function approximates introduced in section IV-A specifically to address these two challenges. Polynomial coefficients can be kept in registers for reuse over many evaluations, and SIMDization is facilitated by the fact that the same polynomial is evaluated for all grid cells. To avoid the complexities of inline assembly, vector intrinsics (or "built-ins") are used to generate SIMD instructions.

There is no coupling between the 12 gate functions, so each has its own separate data stream. Assigning 3 gate functions to each of the 4 hardware threads on a core allows us to store different polynomial coefficients in registers for each thread. This is valuable since each of the four hardware threads per core has its own register bank, corresponding to 4k/core of storage in registers (compared to 16k in L1 cache). The varying polynomial degrees in the rational function approximates complicates the instruction balance within a single thread and impacts load balance, but this is largely remedied by loop skewing and the fact that only two of four threads are required to fully utilize the dual-issue instruction capability; the instruction units can remain heavily utilized even if one or two threads finish early. We found that performance was highly sensitive to loop unrolling and judicious use of unrolling had to be enforced. Greater unrolling causes register spilling, the cost of which is exacerbated by the write-through L1 cache. Register pressure is reduced by storing four different coefficients per register (instead of four copies of the same coefficient) and replicating ("splatting") into operand registers when needed. This technique can also favorably shift the mix of floating point and integer instructions. Finally, we had to bear in mind that our goal was more complicated than obtaining the fastest single gate function, but the fastest computation of all 12 gates. Thus, some routines had to be restricted (e.g., in terms of register-to-register moves or loop unrolling) so that other routines, sharing limited resources, could run more effectively.

Optimization of the non-gate functions is more challenging because of a much greater diversity in the mathematical forms of the equations and because all 20 state variables are needed in the evaluations. With no obviously beneficial way to assign equations to hardware threads, work is instead partitioned

by dividing the grid cells on each core among the threads. This has the unfortunate side effect of limiting the ability to use the per-thread register banks to store the large number of coefficients, as each thread evaluates the same functions. Deep loop unrolling was of limited utility because extreme register pressure when unrolling could result in a catastrophic number of register spills. This was largely ameliorated through an application of the aforementioned technique of splatting coefficients from a table held in registers to operand registers. When employing intrinsics, this work is more painstaking than in the analogous case with assembly, as it is the register coloring and scheduling of the compiler that dictates when this method proves beneficial or pernicious. The third and final hurdle we faced involved the calculation of the log functions. We tailored the polynomial expressions needed to compute our log functions for the known, physically-relevant range of values that we might encounter and exploited SIMD instructions to great advantage.

E. Diffusion calculation

Mathematically, the diffusion calculation is simple. It is only a sum over the 19 stencil points of voltages multiplied by pre-computed coefficients. Due to fiber orientation, these coefficients are different for each grid cell and cannot be reused. The ratio of memory access to arithmetic is thus unfavorably high. Nevertheless, performance does improve with SIMDization obtained by loading the voltages of four adjacent (in memory) grid cells into registers and performing the stencil arithmetic for all four at once. Careful use of register swaps and shifts reduces the number of load operations that are needed. Further reduction in load bandwidth is achieved by representing the diffusion coefficients in single precision. This is reasonable because the fiber direction and conductivity tensor are not known to high precision.

One drawback of our current algorithm is that the grid cells assigned to a node must be represented in memory as a rectangular block. This allows the use of fixed memory offsets to locate stencil points, but requires a copy operation at each time step since reaction performance is optimized by a different memory layout. The fact that not all memory locations in the block correspond to tissue cells creates an additional inefficiency since some SIMD vectors will contain non-tissue cells. This is especially true for nodes assigned to regions near the heart boundary. We expect that improved blocking strategies and optimized assignment of grid cells to nodes will help reduce this problem.

F. Validation

The optimized code has been carefully validated to ensure biological fidelity. Tests of a single cell paced (stimulated) at various prescribed intervals agree very well with published results [13]. We also compare to a TT06 benchmark suite presented in Niederer, et al. [17] involving eleven different electrophysiology codes that use a variety of numerical techniques and computational grids. We find excellent agreement between *Cardioid* and other finite difference codes. The fully anisotropic conductivity tensor was tested using the Method of Manufactured Solutions for a steady state solution of Eq. (1). As expected, the finite volume discretization error is $O(h^2)$ compared with the analytic solution for a grid spacing h.

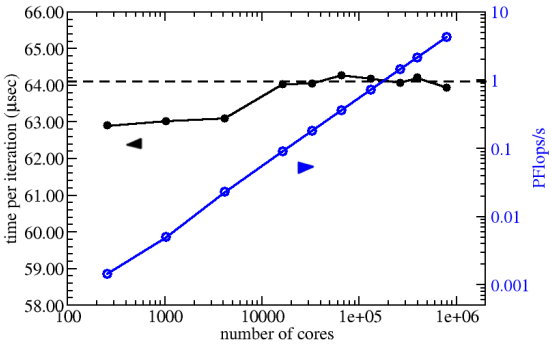

Fig. 3. Weak scaling of uniform block of heart tissue using optimized *Cardioid* code on BG/Q. The time per iteration (in μs) is shown in black, and the throughput (in PFlop/s) is shown in blue.

V. Performance and Scalability

A. Timers and floating point performance counters

We utilize the BG/Q low-overhead CPU time base register to implement low cost *in situ* timers to monitor time spent in different parts of the code. This register, which counts clock cycles, can be accessed via the inline call GetTimeBase() in only 1 cycle.

Each BG/Q core provides 24 counters that can monitor a variety of core events including counting floating point operations. Converting counts to Flop/s requires multiplication by appropriate weights: 1 for addition, subtraction, multiplication, and other non-storage operations, 2 for fused multiply-add, and 8 for division and square root. The corresponding SIMD instructions have four times the weight, i.e., 8 for a quad FMA. Using the IBM Hardware Performance Monitoring (HPM) library, operation counts are summed across all the cores on all nodes and divided by the elapsed time to get the throughput in PFlop/s. Measured HPM throughout values have been verified by directly counting operations using a small test system.

B. Performance and scaling

In order to estimate full machine performance for our target 0.1 mm human heart system, we carry out weak scaling tests on a tissue block having 3584 grid cells per node. The results are shown in Figure 3. We observe excellent weak scaling from 16,384 cores all the way up to 786,432 cores, one half of the full machine, with the time per iteration remaining constant and the measured floating point performance increasing linearly with the number of cores. This confirms that communication costs do not grow with increasing problem size, and that the code is capable of operating efficiently at very large task counts. It is therefore not unreasonable to extrapolate these curves another factor of two to the full

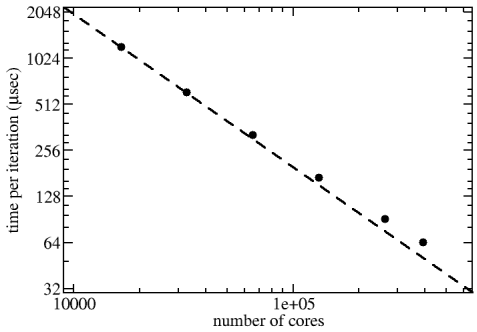

Fig. 4. Strong scaling of uniform block of heart tissue. The dashed line represents perfect scaling.

machine and expect excellent parallel efficiency at full scale. Note too that on the half-machine our throughput exceeds 4.3 PFlop/s, which is over 43% of peak. Combined with our perfect weak scaling, we should exceed 8.6 PFlop/s on the full Sequoia once it is available.

Figure 4 shows the strong scaling performance with a block of approximately 88 million grid cells using up to 393,216 cores (one quarter of the full machine). The speedup from 16,384 cores to 393,216 was 19.0, which corresponds to a parallel efficiency of 79%. As expected, there is some loss of efficiency when going from nearly 100,000 grid cells per node down to 3584: although the 3584 cell/node case is highly optimized (as shown by the floating point performance), the operating margins at such light workloads are so narrow that normally inconsequential costs such as loop startup and subprogram calls become significant. It should be noted that were we merely interested in achieving high flop rates, it would have been easy to simply construct systems with larger local data sizes. For example, our 16384-core case runs at over 54% of peak. However, our goal is ultimately to optimize the time to solution of a real scientific application, thus necessitating the far more difficult task of optimizing performance at the far end of the strong scaling curve.

We next compare the optimized *Cardioid* with the unoptimized version (based on reaction code fom the CellML Model repository and a simplistic implementation of the diffusion operator). The optimized code exhibits much better weak scaling than the unoptimized version (which uses MPI and OpenMP). On average, the optimized code executes about 50 times faster than the unoptimized code.

In addition to the performance trials just described, we have confirmed the stability of the code with an initial 20 heartbeat run using a 180 million grid cell heart geometry on 16,384 BG/Q nodes. The additional geometric complexity of the heart structure does lower performance. This is almost entirely due to the inefficiencies in the diffusion implementation mentioned in section IV-E that affect odd-shaped subdomains with a large number of non-tissue grid cells. Improvements to the diffusion and subdomain assignment algorithms are in progress

TABLE I
THROUGHPUT (HEARTBEATS PER WALL CLOCK MINUTE) OF CARDIAC ELECTROPHYSIOLOGY MODELS TARGETED TO HPC SYSTEMS.

Author	Resolution	Grid cells	Raw thrpt	Norm. thrpt
Pope	0.2mm	32.5M	0.027	0.0024
Niederer	0.25mm	26M	0.025	0.018
Krause	0.0625mm	1560M	0.0015	0.0065
Mirin	0.1mm	370M	9.0 (est.)	9.0 (est.)

to remedy this issue. The heart geometry also differs from the block trial in that it contains multiple cell types. Since different cell types use different parameters in the reaction model, this could have a performance impact; however this potential problem is easily avoided by sorting the cells by type during initialization.

With algorithmic improvements and additional refinements to our optimization strategy, we expect to obtain performance on heart geometries that is nearly-comparable to that of a homgeneous tissue block. We should therefore achieve our target throughput of 8–10 heartbeats per minute on our 370 million grid cell heart on the full system.

C. Time-to-solution

We compare the *Cardioid* time to solution with those of other leading cardiac electrophysiological models. We were able to identify three such efforts, all having published or presented in 2011, namely [10], [4] and [11]. All three invoke the monodomain approximation and utilize a reaction-diffusion model based on either [13] or [18]. All utilize an explicit time integrator for the reaction phase, and two of the three offer an implicit option for the diffusion phase. The quoted performance figures pertain to different sized hearts and are largely exclusive of I/O. Results are shown in Table I.

The raw throughput is defined as the number of simulated heartbeats per wall minute. In order to compare codes, we linearly scale raw throughput to 370M, the number of tissue grid cells for the study herein, to get normalized throughput, which appears in the rightmost column of the table. We believe that the published time steps should apply at the normalized resolution, so we do not apply a timestep scaling. We see that the closest competitor to *Cardioid* is that of Niederer, whose time-to-solution is 500 times longer, and that the Krause effort is 3 times slower still.

VI. SCIENTIFIC CAPABLITY AND RESULTS

A. Anatomical reconstruction

We reconstruct anatomical models of the human ventricles from cryosectional images (see Figure 5) from the Visible Human Project® [7] of The National Library of Medicine. The female dataset is used since the cryosectional images have higher spatial resolution (0.33x0.33x0.33 mm) than those of the corresponding male dataset. A full description of this procedure is beyond the scope of this article. Briefly, the 2D images are segmented at organ boundaries and stacked to form a 3D structure from which is defined a 3D finite element mesh. We apply an additional transformation to define the Cartesian

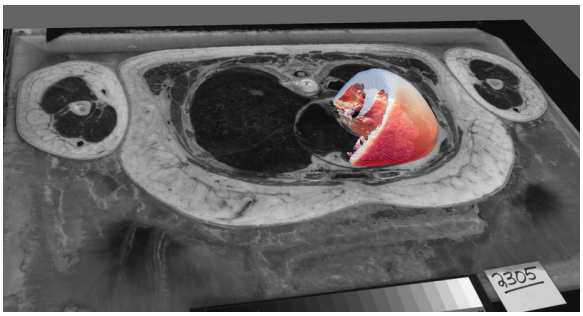

Fig. 5. Anatomical reconstruction of the 3D heart mesh from 2D images of the Visible Human Project.

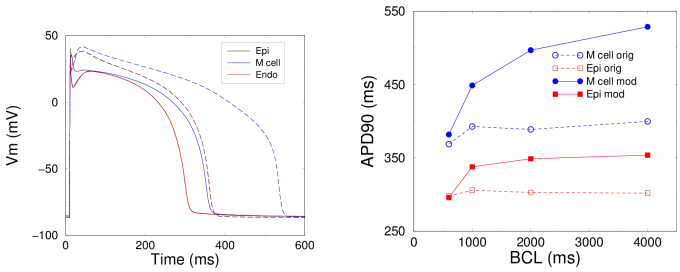

Fig. 6. Electrophysiological response in epi, M and endo cells. (Left): AP's are shown for different cell types, labeled according to color. The responses are shown for 600 ms BCL (solid) and 4000 ms BCL (dashed). Note the epi and endo traces mostly overlap except for the notch after the rapid upstroke. (Right): The action potential duration at 90% polarization (APD90) is shown as a function of pacing rate for epi and M cells for the default (dashed) and modified (solid) TT06 models. The data for endo (not shown) are very similar to that for epi.

rectangular grid that is used in the code. To compute the conductivity for the diffusion calculation, we develop novel methods to define fibers, even within the smallest structures that were not resolvable with previous methods. For example, the complex surface geometry on the endocardium (inner layer of the ventricles; see Figure 1) was not accounted for in previous models of human hearts. Our resulting grids contain 370M (0.1 mm), 180M (0.13 mm), 89M (0.16 mm), and 44M grid cells (0.2 mm).

B. Impact of extension of reaction model

We examine the impact of our reaction model extensions on the various cell types — epi, endo and M. The three cell types have similar action potential duration (APD), the width of the spike in cell membrane potential after triggering, but the epi cell has a more pronounced notch after the initial upstroke (see Figure 6, left). Based on studies of human M cell ventricular wedges [19], [20], the APD is greatly prolonged with longer basic cycle length (BCL), the time between trigger events. The TT06 model, however, shows little dependence of APD on cycle length, a deficiency noted in the original papers and attributed to lack of a late sodium current in these models. With our addition of such a current, which is preferentially larger in M cells as characterized by experimental studies [21], [22], we are able to demonstrate a more realistic dependence of APD on basic cycle length (see Figure 6, right).

C. Wedge simulations

Torsade de Pointe (TdP) is an important class of arrhythmias that manifest as rapid undulations in the ECG with an envelope function that varies more slowly over time. The proposed conceptual models for the formation of TdP [2], [23] involve two key components: heterogeneity of APD at different tissue locations and a triggering event that initiates the reentrant activation. The heterogeneity of APD is thought to arise from the M cell population that shows preferential APD prolongation in response to slower heart rates and also to Class III anti-arrhythmic drugs like d-sotalol. The trigger events are generally assumed to result from extra excitation events such as early-afterdepolarizations (EAD's), in which a cardiac cell that is repolarizing abruptly depolarizes with a secondary and smaller upstroke.

We perform wedge simulations that mirror experimental studies that show that slow pacing rates and d-sotalol work synergistically to promote TdP [24]. To investigate Tdp formation, we simulate a wedge of 32x35x32 mm, roughly the size of experimental preparations, that contains approximately 3M grid cells that are extracted from the whole heart. M cells, generally thought to be located in the mid-myocardium, are more recently described as existing in island-like subdomains in the ventricular wall, though reports vary for different species and regions of the heart [6]. We choose to place the M cells in islands occupying roughly 30% of the volume of the wall [19] and located more toward the endocardium as described recently in human wedges [20]. We use a slow pacing rate of 2000 ms BCL that prolongs the APD of M cells to a greater extent than with other cell types (Figure 6). The mode of action of d-sotalol is to block a particular class of channels that promote repolarization, with a net effect to increase the APD. We set the corresponding current to 0 to simulate the blockage of these channels.

Representative simulation results are shown in Figure 7, in which the wedge is stimulated on the endocardial surface (left side) at 0 ms by the first stimulus (termed S1). The activation wavefront propagates in the normal direction from the endocardium to the epicardium, reaching roughly halfway by 50 ms. After the activation front passes, the cells begin to repolarize toward rest, but the M cell population remains depolarized longer, as seen by the islands on the lower left side of the wedge at 560 ms. A second stimulus (termed S2) is triggered on the upper quadrant of the endocardial surface to simulate an extra stimulus from the Purkinje fiber network [25], [26]. The second wavefront begins to propagate in a clockwise direction around the functional block created by M cells that have yet to repolarize and therefore cannot be excited again. In the case with the drug (left), the reentrant pattern continues for the complete duration of the 2000 ms simulation. In comparison, without the drug the smaller effective block by the island does not support sustained reentry, and the activation of the tissue ceases by 690 ms, enabling the

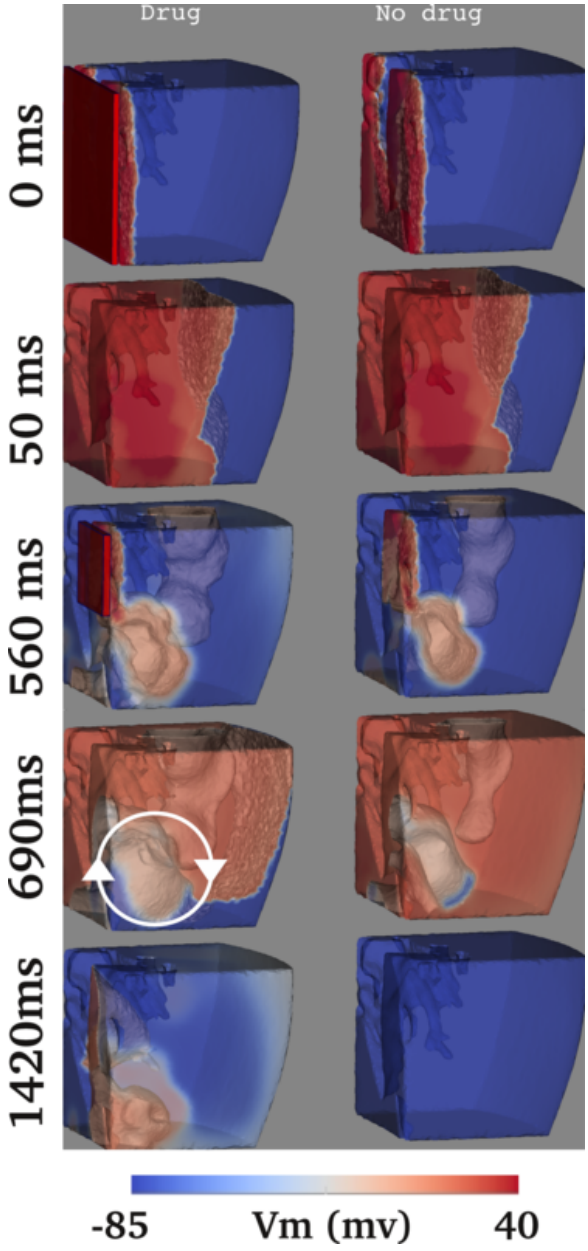

Fig. 7. Activation pattern in a wedge model. The color shows the membrane potential as indicated by the scale bar. On the left, labels show the time from the initial stimulus (S1) applied at the location of the red block in the uppermost panel at 0 ms. A second stimulus (S2) is applied at 560 ms at the location shown by the smaller red block. The left column shows a simulated drug response where reentrant activation occurs, illustrated symbolically by the rotating figure. The right column shows a control case with no drug, and reentrant activation does not occur.

tissue to fully repolarize.

D. Whole ventricle and ECG results

Figure 8 shows results for the two ventricle model with an activation (depolarization) and relaxation (repolarization) sequence that would occur in a typical heartbeat. Here the grid has 0.16 mm resolution and consists of 90M grid cells representing heart tissue. Electrical stimuli are applied at points near the endocardial surface to simulate actual activation of

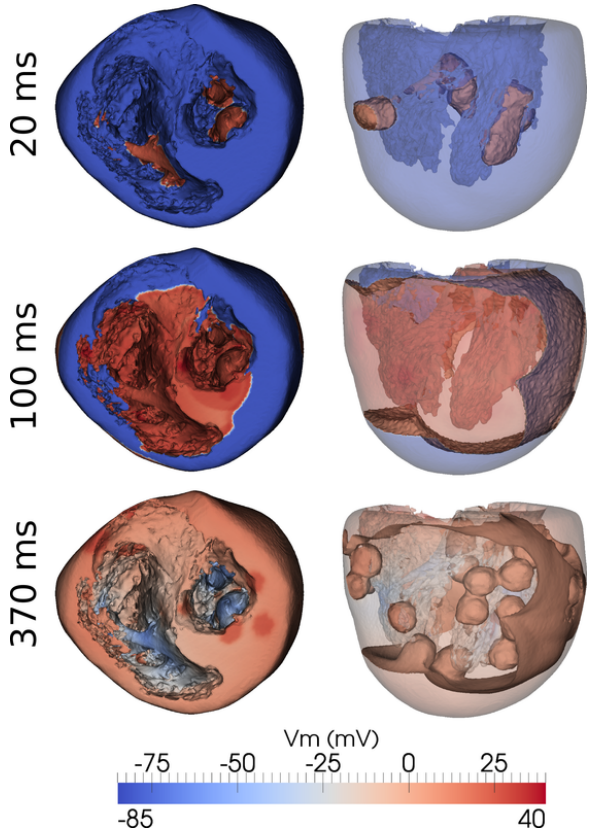

Fig. 8. Activation and relaxation pattern in the full two ventricle model with the drug application. A top-down view is shown in the left column, and a front view is shown in the right column (the right ventricle appears on the left side). The color indicates the cellular potential as shown by the scale bar. The front of electrical activation starts at the endocardium (20 ms) and propagates towards the epicardium (100 ms). During repolarization of the ventricles (370 ms), spherically shaped M-cell islands are clearly seen because M cells repolarize later than the surrounding tissue.

the cardiac ventricles by Purkinje fibers. The locations of the pacing sites are chosen to match the early activation sites mapped in seminal work of [27]. At 20 ms, the activation wave front is still localized to the region near the initial stimuli. At 100 ms, the activation wave front reaches the outer edge of the epicardial surface throughout most of the ventricles, while only the base (top of heart) and the apex (bottom of heart) remain in a relaxed state, as indicated by the small blue regions. The activation eventually spreads to the whole heart, and then repolarization occurs as the heart cells transition back to the resting potential. The first areas that relax and revert back to blue are seen in the apex of the heart toward the endocardial surface at 370 ms. In the side view at 370 ms, one can also see repolarization wavefronts where the central region has mostly relaxed and the wave of repolarization is spreading toward the base and apex. However, there are also clearly demarcated M-cell islands that arise because the M cells repolarize later than the surrounding tissue. These remain after the main wave of repolarization has passed.

One goal of the simulations is to produce results that can be compared to existing experimental data. To illustrate this

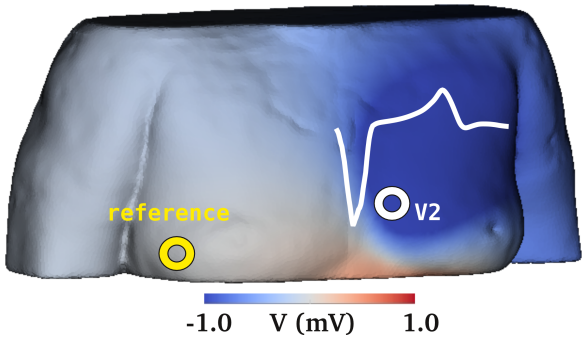

Fig. 9. Electrical potential of the torso and the simulated ECG. The body surface potentials at 70 ms are mapped as colors shown on the scale bar. The simulated ECG, shown by the white trace, is recorded at the V2 precordial electrodes of a standard 12-lead configuration.

capability, we generate an ECG in which the changes in the membrane potentials in the two ventricle model result in changes of the electrical potentials on the body surface. The body surface potentials are computed by solving the static diffusion equations over the finite element model with the geometry of the real torso. These potentials are mapped as colors on the surface of the reconstructed torso in Figure 9. In a clinical setting, the body surface potentials are recorded at discrete electrode positions to produce time traces. We show an example ECG time trace recorded from the precordial lead V2 of a 12-lead ECG. The computed ECG time trace is similar to that of clinical recordings for a normal heartbeat.

One key question to address is whether the reentrant arrhythmias demonstrated in wedges can be also demonstrated in the two ventricle model. The follow-on question is whether the resulting ECG will have the characteristic undulating pattern that is observed for TdP in the clinical setting. Because the model executes quickly, we can explore parameter space with long runtimes — capabilities well beyond those of previous models.

VII. Conclusions

We have developed a highly efficient and scalable simulation capability for human ventricular electrophysiology with groundbreaking detail and resolution to provide new insights into arrhythmia initiation and maintenance. The code solves a reaction-diffusion equation for the cell membrane potential using a standard ionic model of the human cell with novel modifications to better represent heterogeneous electrical properties of different cell types. We use an anatomical representation of the ventricles that, at 0.1 mm resolution, is best in its class for organ-level studies of arrhythmogenesis.

The code is optimized for the Sequoia BG/Q machine at Lawrence Livermore National Laboratory. The optimization approach combines biology-aware and hardware-aware tuning and utilizes both distributed and shared memory parallelism. The 64 hardware threads on a node are allocated to the reaction, diffusion and communication phases to maximize utilization of the hardware. We have demonstrated near-perfect weak scaling for a block of tissue cells using a range of processors representing 1/96 to 1/2 of the full machine, as well as excellent strong scaling. The optimized code is a factor of 50 faster than the unoptimized starting point. Despite operating in an aggressive strong scaling regime with limited computational work per core, our scaling studies predict performance of 8.6 PFlop/s (43% of peak), and absolute throughput of roughly 9 heartbeats per minute (less than 65 μs per time step), for a 0.1 mm resolution 370 million cell grid on the full Sequoia. The corresponding time-to-solution is 400–500 times faster than the state-of-the-art.

The results of the scientific studies illustrate the power of the modeling approach. Consistent with experiments, the model shows that long action potentials of the M cell islands, accentuated by slow heart rates and d-sotalol, produce regions of conduction block that facilitate the formation of reentrant activation patterns. The model and its efficient implementation greatly advance the state-of-the-art and already provide an unparalleled method for interpreting and predicting electrophysiological responses that span from wedges to ventricles and to near-cellular scale. For example, we plan to simulate experimental drug protocols using real patient data that show the spontaneous formation and termination of arrhythmias in ECG records collected over many hours. Moreover, our method has the potential to make practical new cardiac modeling applications such as drug safety testing, medical device design, and therapeutic planning for personalized medicine. Once Sequoia installation is complete, we will have the opportunity to carry out our planned multi-hour drug simulations, which we aim to present at SC12.

Acknowledgements

We thank Michel McCoy and the NNSA Advanced Scientific Computing program, as well as the Institute for Scientifc Computing Research for their support of this effort. We also thank Frederick Streitz, James Sexton and Ajay Royyuru for their expert guidance. We additionally thank Alexandre Eichenberger, Blake Fitch, Fabrizio Petrini, Mike Pitman, Simon Wail and Bob Walkup of IBM Research for their contributions to the cardiac code and its optimization and Ruud Haring, Martin Ohmacht, and Rob Shearer for information on the BG/Q architecture. We further thank Kim Cupps, David Fox, Adam Bertsch, and the Livermore Computing team for working closely with us to assure the readiness of the hardware and software and their advice on its use.

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. This is LLNL report LLNL-CONF-553873.

References

[1] N. A. Estes, 3rd, "Predicting and preventing sudden cardiac death," *Circulation*, vol. 124, no. 5, pp. 651–656, 2011.

[2] D. P. Zipes and H. J. Wellens, "Sudden cardiac death," *Circulation*, vol. 98, no. 21, pp. 2334–2351, 1998.

[3] J. D. Moreno and Z. I. Zhu, et al., "A computational model to predict the effects of class I anti-arrhythmic drugs on ventricular rhythms," *Sci. Transl. Med.*, vol. 3, no. 98, p. 98ra83, 2011.

[4] S. Niederer, L. Mitchell, N. Smith, and G. Plank, "Simulating human cardiac electrophysiology on clinical time-scales," *Frontiers in Physiology*, vol. 2, no. 14, pp. 1–7, 2011.

[5] M. Reumann, B. G. Fitch, A. Rayshubskiy, M. C. Pitman, and J. J. Rice, "Orthogonal recursive bisection as data decomposition strategy for massively parallel cardiac simulations," *Biomed Tech*, vol. 56, no. 3, pp. 129–145, 2011.

[6] D. U. Keller and D. L. Weiss, et al., "Influence of I(Ks) heterogeneities on the genesis of the T-wave: a computational evaluation," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 2, pp. 311–322, 2012.

[7] U.S. National Library of Medicine Visible Human Project. http://www.nlm.nih.gov/research/visible/visible_human.html.

[8] M. K. Das and D. P. Zipes, "Antiarrhythmic and nonantiarrhythmic drugs for sudden cardiac death prevention," *J. Cardiovasc. Pharmacol.*, vol. 55, no. 5, pp. 438–449, 2010.

[9] N. Sarapa and J. Morganroth, et al., "Electrocardiographic identification of drug-induced QT prolongation: assessment by different recording and measurement methods," *Ann. Noninvasive Electrocardiol.*, vol. 9, no. 1, pp. 48–57, 2004.

[10] B. J. Pope, B. G. Fitch, M. C. Pitman, J. J. Rice, and M. Reumann, "Performance of hybrid programming models for multiscale cardiac simulations: Preparing for petascale computation," *IEEE Trans. Biomed. Eng.*, vol. 58, pp. 2965–2969, 2011.

[11] D. Krause, M. Potse, T. Dickopf, R. Krause, A. Auricchio, and F. Prinzen, "Hybrid parallelization of a realistic heart model," in *Proceedings of Supercomputing 2011*. Los Alamitos, CA: IEEE Computer Society Press, 2011.

[12] M. Potse, B. Dube, J. Richer, A. Vinet, and R. Gulrajani, "A comparison of monodomain and bidomain reaction-diffusion models for action potential propagation in the human heart," *IEEE Trans. Biomedical Engineering*, vol. 53, no. 12, pp. 2425–2435, 2006.

[13] K. H. W. J. ten Tusscher and A. V. Panfilov, "Alternans and spiral breakup in a human ventricular tissue model," *American Journal of Physiology - Heart and Circulatory Physiology*, vol. 291, pp. 1088–1100, 2006.

[14] S. Rush and H. Larsen, "A practical algorithm for solving dynamic membrane equations," *IEEE Trans. Biomed. Eng.*, vol. 25, pp. 389–392, 1978.

[15] R. Koradi, M. Billeter, and P. Guntert, "Point-centered domain decomposition for parallel molecular dynamics simulation," *Comput. Phys. Commun.*, vol. 124, pp. 139–147, 2000.

[16] R. A. Haring, M. Ohmacht, T. W. Fox, M. K. Gschwind, D. L. Satterfield, K. Sugavanam, P. W. Coteus, P. Heidelberger, M. A. Blumrich, R. W. Wisniewski, A. Gara, G. L.-T. Chiu, P. A. Boyle, N. H. Christ, and C. Kim, "The ibm blue gene/q compute chip," *IEEE Micro*, vol. 32, no. 2, pp. 48–60, 2012.

[17] S. Niederer, E. Kerfoot, A. P. Benson, M. O. Bernabeu, O. Bernus, C. Bradley, E. M. Cherry, R. Clayton, F. H. Fenton, A. Garny, E. Heidenreich, S. Land, M. Maleckar, P. Pathmanathon, G. Plank, J. F. Rodriguez, I. Roy, F. B. Sachse, G. Seemann, O. Skavhaug, and N. P. Smith, "Verification of cardiac tissue electrophysiology simulators using an n-version benchmark," *Phil. Trans. R. Soc. A*, vol. 369, pp. 4331–4351, 2011.

[18] K. H. W. J. ten Tusscher, D. Noble, P. J. Noble, and A. V. Panfilov, "A model for human ventricular tissue," *American Journal of Physiology - Heart and Circulatory Physiology*, vol. 286, pp. 1573–1589, 2004.

[19] E. Drouin and F. Charpentier, et al., "Electrophysiologic characteristics of cells spanning the left ventricular wall of human heart: evidence for presence of m cells," *J. Am. Coll. Cardiol.*, vol. 26, no. 1, pp. 185–192, 1995.

[20] A. V. Glukhov, V. V. Fedorov, Q. Lou, V. K. Ravikuman, P. W. Kalish, R. B. Schuessler, N. Moazami, and I. R. Efimov, "Transmural dispersion of repolarization in failing and nonfailing human ventricle," *Journal of the American Heart Association*, vol. 106, pp. 981–991, 2010.

[21] C.Antzelevitch and W. Shimizu, et al., "The m cell: its contribution to the ecg and to normal and abnormal electrical function of the heart," *J. Cardiovasc. Electrophysiol.*, vol. 10, no. 8, pp. 1124–1152, 1999.

[22] A. C. Zygmunt and G. T. Eddlestone, et al., "Larger late sodium conductance in m cells contributes to electrical heterogeneity in canine ventricle," *Am. J. Physiol. Heart Circ. Physiol.*, vol. 281, no. 2, pp. H689–697, 2001.

[23] C. Antzelevitch and A. Burashnikov, "Overview of basic mechanisms of cardiac arrhythmia," *Card. Electrophisiol. Clin.*, vol. 3, no. 1, pp. 23–45, 2011.

[24] F. G. Akar and G. X. Yan, et al., "Unique topographical distribution of m cells underlies reentrant mechanism of torsade de pointes in the long-qt syndrome," *Circulation*, vol. 105, no. 10, pp. 1247–1253, 2002.

[25] M. Monserrat and J. Saiz, et al., "Ectopic activity in ventricular cells induced by early afterdepolarizations developed in purkinje cells," *Ann. Biomed. Eng.*, vol. 28, no. 11, pp. 1343–1351, 2000.

[26] M. Deo and P. M. Boyle, et al., "Arrhythmogenesis by single ectopic beats originating in the purkinje system," *Am. J. Physiol. Heart Circ. Physiol.*, vol. 299, no. 4, pp. H1002–1011, 2010.

[27] D. Durrer and R. T. van Dam, et al., "Total excitation of the isolated human heart," *Circulation*, vol. 41, no. 6, pp. 899–912, 1970.