Request for Research Allocation on SuperMIC: Simulating Radiation Damage to Vascular Networks

William P. Donahue (user:wdonah2) & Wayne D. Newhauser (user: newhaus)

1 Problem Statement

We are requesting 3,000,000 service units (SUs) for the final optimization and timing measurements of a scalable algorithm to simulate radiation damage to whole organ vascular networks on the LSUHPC SuperMIC cluster. Almost 2/3$^{\text{rds}}$ of all cancer patients receive radiotherapy as part of their treatment (Pasko et al., 2000). As cancer treatment has improved through the years (Siegel et al., 2017), increasing importance must be placed on understanding the biologic mechanisms of radiation late effects. Radiation damage to the vasculature of healthy tissues has been associated with multiple life-threatening effects including white-matter necrosis in the brain (Lyubimova and Hopewell, 2004), cardiac disease (Stewart et al., 2010), and lung fibrosis (Yarnold and Brotons, 2010). While previous research has used laboratory experiments and small animal models, these are fraught with uncertainties and are expensive. Computational models of the radiation damage to vasculature could provide an efficient platform for performing exploratory research before investing in larger experiments.

The goal of this work is to simulate the biological effect of radiation damage on the vasculature. While the simulation of radiation transport is a mature field (Newhauser and Zhang, 2015), it is typically limited to volumes much greater than that of an individual blood vessel.The human brain contains approximately 3-10 billion vessels with dimensions from centimeters to micrometers (Gray and Lewis, 1918; Linninger et al., 2013; Lauwers et al., 2008). This proposal supports the final tuning and timing measurements of our dose calculation algorithm. Additionally, we will complete an end-to-end test of our damage simulation framework from vascular geometry definition to the simulation of radiation-induced changes in blood flow. This work is the final part of a more extensive dissertation project with the goal to demonstrate the computational feasibility of simulating the impact of radiation-induced damage on vascular networks.

More details of this dissertation project are provided as an attachment.

2 Background

The first step in determining the damage radiation causes is calculating the energy deposited into the tissue. In radiation therapy, the two primary methods for performing this calculation are Monte-Carlo simulation and analytical modeling (Newhauser et al., 2017). Due to the stochastic nature of radiation interactions in matter, analytical methods can only be used on larger targets (*i.e.* millimeter voxels of tissue). Monte Carlo methods, on the other hand, simulate the multi-scale nature of radiation energy deposit on a track by track basis. This creates an accurate description of the deposited dose, but is extremely time-consuming. Improvements in computational power have allowed for more accurate calculations in many fields, including radiation oncology and medical physics.

Using a Startup Allocation on SuperMIC (hpc_wd_vdc_dev), we developed an algorithm to probe the computational feasibility of performing detailed dose calculations on whole-organ vascular networks. This algorithm uses simplified track-structure models and vascular geometry. The previous allocation was used to collect preliminary timing and scalability results. The timing data was used to create a model of the computation time versus four simulation parameters. This model will be discussed further in Section 4. While the model suggests it is currently not feasible to simulate radiation dose with a high statistical uncertainty to the human brain, it does appear feasible in a rodent brain. This is the primary focus of the work to be completed in this allocation.

3 Methodology

All the applications used in our framework were developed in-house. They used multiple libraries to support various functions.

All software and libraries used in this study will be compiled with the Intel C++ Compiler (2018, Intel Corporation, Santa Clara, CA, United States). BLAS, LAPACK, and other math routines, are provided by the Intel Math Kernel Library (MKL) (2018, Intel Corporation, Santa Clara, CA, United States). Finally, we use the Intel MPI Library (2018, Intel Corporation, Santa Clara, CA, United States) for all distributed memory communication.

All input and output (I/O) operations for the applications utilize the Hierarchical Data Format version 5 (HDF5) library (Folk et al., 1999). The application leverages the parallel I/O capabilities of the library and the filesystem to improve data throughput. The library is tuned to the local file system following the suggestions of Howison et al. (2012).

Graph partitioning in the Network Preprocessing application is computed with the parallel METIS algorithm (ParMETIS) (Karypis and Kumar, 1998; Karypis, 1999). We use the K-way partitioning routine on an edge graph of our vascular network. The ParMETIS library must be compiled with support for large indexes and double precision real values, to handle networks that have billions of vessels.

Matrix solving is performed using the Hypre Solver Library (Falgout and Yang, 2002). We will compile the library with support for OpenMP, MKL, and large (64-bit) indexes. This library is published and maintained by Lawrence Livermore National Laboratory (LLNL).

4 Requirements Analysis

4.1 Computational Speed

To assist in the determination of computational feasibility we have created an equation of the computation time versus four important parameters of the simulation. These parameters are: the number of particles simulated (N_p), the number of vessels in the simulation (N_v), the initial energy of the particles (E), and a correction factor for the geometric distribution of vessels (FS). The general equation for this model is

$$T = T_0 \cdot P(N_p, N_v) \cdot V(N_v) \cdot F(E) \cdot H(FS) \tag{1}$$

where T is the computation time in CPU-Hours, T_0 is a time to simulate a baseline set of parameters, and $P(N_p)$, $V(N_v)$, $F(E, N_v)$, and $D(FS)$ are scaling factors for each of the four parameters. The baseline case took approximately 27 minutes to compute the dose to 2 million vessels from 100,000 protons with a initial energy of 100 MeV, with a density correction factor of 2. The algorithm for dose computation is parallelized on the number of vessels in the network and the dose computation is independent on each compute node. Figure 1 plots the four scaling functions. Time is dominated by the number of particles that are simulated, which is also the driving factor in the statistical accuracy of the simulation. Currently the model predicts the training data within 25% of the original value for all parameters.

Table 1 shows the calculated timing results for various size models to be studied using this allocation. The most time consuming models will be those of mouse and human brain. Other models listed here will be used for validation of the timing model. Each validation run will be run at least 5 times, while the networks representing the mouse and human brains will be run at most three times. The stretch goal is the ideal outcome for the dissertation project. If possible we would like to run this model once. The algorithm is designed to use successive runs to accumulate the dose from the requisite number of particles.

Table 1: This table contains predicted number of SUs for some of the models to be run using this allocation. The onse labeled as mouse, child, adult brains, are the most time consuming.

Model Name	Number of Vessels	Number of Particles	Energy (MeV)	Density Corr. (%)	SUs	Time (h)	Number of Nodes
Validation 1	126	100	0.1	0.0	7.3e-09	3.7e-10	1
Validation 2	2046	1000	250.0	3.0	0.00037	1.9e-05	1
Validation 3	1048574	1000000	50.0	0.0059	1.0	0.051	1
Validation 4	1073741822	100000	20.0	100.0	460.0	23.0	1
Validation 5	2097150	100000000	150.0	0.003	540.0	3.4	8
Mouse 1	4194302	1000000000	90.0	4.8e-05	2600.0	1.0	128
Mouse 2	4194302	10000000	90.0	4.8e-05	26.0	0.02	64
Child Brain	2147483646	1000000	170.0	0.0	390.0	0.15	128
Adult Brain	4294967294	1000000	170.0	0.0	910.0	0.35	128
Adult Brain (Stretch Goal)	4294967294	1000000000	170.0	0.0	920000.0	360.0	128

4.2 Memory and Storage Requirements

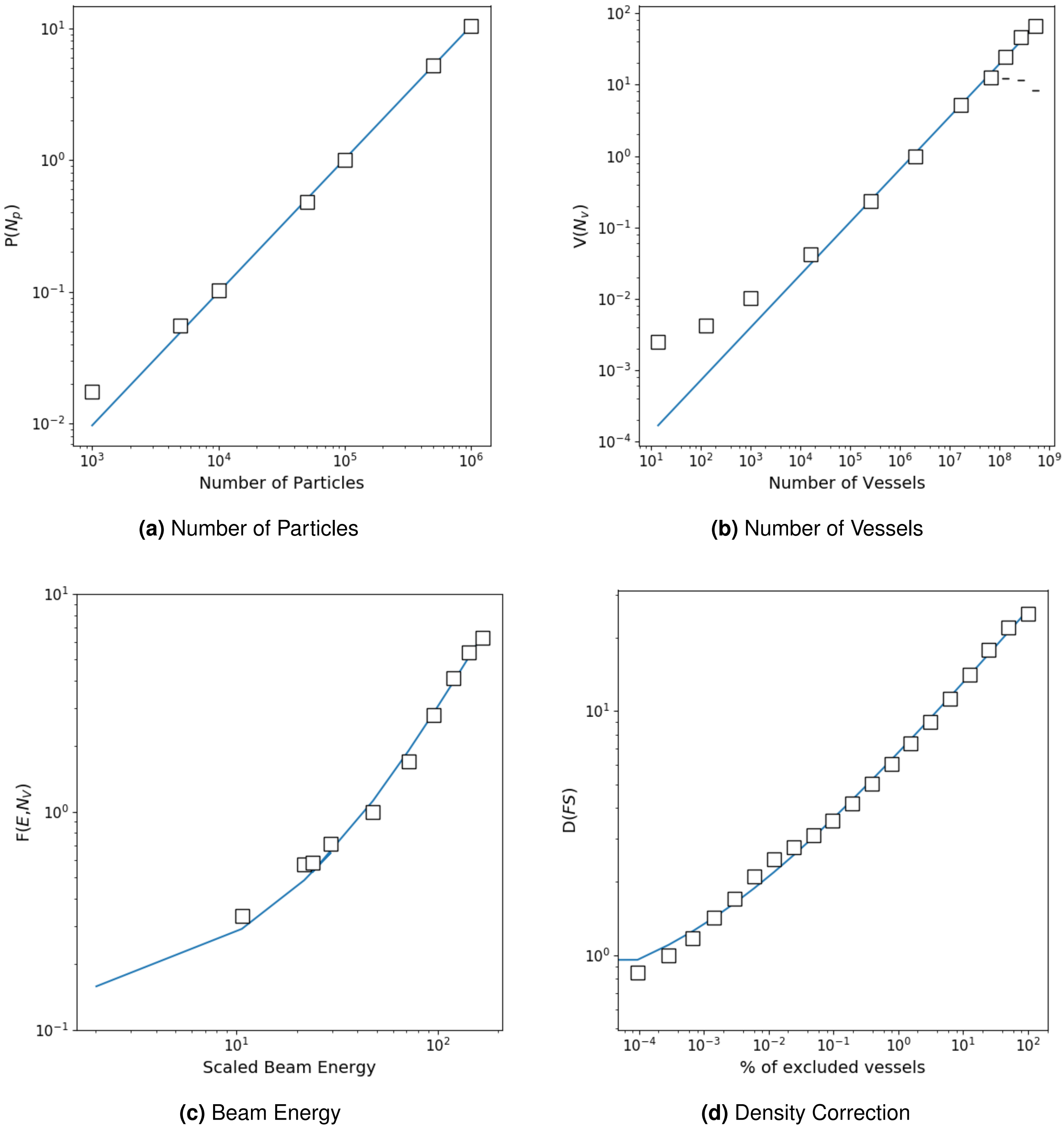
The memory requirements of the dose calculation algorithm are less than 30 GB by design, it is primarily latency and compute-bound. The blood calculation algorithms require approximately 64 GB of memory during the computation. These will only be run once during the end-to-end testing.

The storage requirements are quite large. Table 2 shows the storage requirements for the network sizes to be used in this allocation.

Table 2: Contains predicted number of GBs needed on disk for the models in Table 1. The total is broken into 3 different components demonstrating that the dose data requires the most storage.

Model Name	Number of Vessels	Geometry Storage (GB)	Dose Storage (GB)	Blood Flow Storage (GB)	Total GBs
Validation 1	126	1.03e-05	0.00015	1.88e-06	0.000162
Validation 2	2046	0.000168	0.00244	3.05e-05	0.00264
Validation 3	1048574	0.0859	1.25	0.0156	1.35
Validation 4	1073741822	88.0	1280.0	16.0	1380.0
Validation 5	2097150	0.172	2.5	0.0312	2.7
Mouse 1	4194302	0.344	5.0	0.0625	5.41
Mouse 2	4194302	0.344	5.0	0.0625	5.41
Child Brain	2147483646	176.0	2560.0	32.0	2770.0
Adult Brain	4294967294	352.0	5120.0	64.0	5540.0
Adult Brain (Stretch Goal)	4294967294	352.0	5120.0	64.0	5540.0

These models are transient, and will only be on the cluster for the duration of calculations. Each



(a) Number of Particles

(b) Number of Vessels

(c) Beam Energy

(d) Density Correction

Figure 1: Plots of the 4 models used to calculate scaling factors.

will be immediately removed from the cluster upon completion of testing.

4.3 Conclusions of Requirements Analysis

The computation requirements listed above demonstrate the computational needs of our algorithm. The most intensive trials should be our modeling of mouse and human brains. We still have some development to perform on algorithm to improve the load balancing across nodes. We predict that this will take about 200,000 SUs. Due to uncerstainties involved in software development, we are requesting an additional 200,000 SUs for this phase.

Following this, we perform final data collection for the training and validation data of our timing model. This should require approximately 200,000 SUs, as we expect to try additional validation points. We request an addition 100,000 SUs, in case our timing model requires additional data for training, validation, and diagnosis of errors and deviations.

Finally, we are requesting 2,300,000 SUs for calculation of dose to the Mouse and Human Brains. This number includes computation time to perform our stretch goal computation, which consumes about half of this request. Additional time (300,000 SUs) in this request is dedicated the computation of blood flow and other utility applications needed for our end-to-end tests.

This results in the following requested resources.

Application Development	400,000 SUs
Training and Validation Data	300,000 SUs
Simulation of Whole Brain Dose	2,300,000 SUs
Total Requested SUs	**3,000,000 SUs**

5 Research Schedule

Immediately upon assignment of the allocation, tuning will begin on the dose calculation algorithm. This should be completed within the first month. All model computations will be completed in the spring semester. This allocation, and the work described in this proposal represent the final measurements of a PhD research project, which will be defended by May 11$^{\text{th}}$, 2019.

Any remaining service units will be used to support other active research projects in the Newhauser Research Group and the LSU Medical Physics and Health Physics Graduate Program. One example is extending the framework to simulate the impact of large projectiles on the vascular network.

6 Grant Support and Publications

The current funding for this work is from the Bella Bowman Foundation. Its principle purpose is to fund a graduate student fellowship.

The work proposed here will be the focus of two publications and a dissertation. Additionally, data collected during this study will be used as preliminary results for grant applications to the NIH, DARPA, and NSF.

References

Robert D. Falgout and Ulrike Meier Yang. hypre: A Library of High Performance Preconditioners. In *Computational Science ICCS 2002.*, pages 632–641. Springer, Berlin, Heidelberg, 2002. ISBN 978-3-540-43594-5. doi: 10.1007/3-540-47789-6_66.

Mike Folk, Albert Cheng, and Kim Yates. HDF5: A file format and I/O library for high performance computing applications. In *Proceedings of supercomputing*, volume 99, pages 5–33, 1999.

H Gray and W H Lewis. *Anatomy of the Human Body.* Lea & Febiger, 1918.

Mark Howison, Quincey Koziol, David Knaak, John Mainzer, and John Shalf. Tuning HDF5 for Lustre file systems. *IASDS '10 Proceedings of the Workshop on Interfaces and Abstractions for Scientific Data Storage*, 5, 2012.

George Karypis. Parallel Multilevel k -Way Partitioning Scheme for Irregular Graphs. *SIAM Review*, 41 (2):278–300, 1999. ISSN 0036-1445. doi: 10.1109/SUPERC.1996.183537.

George Karypis and Vipin Kumar. A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, jan 1998. ISSN 1064-8275. doi: 10.1137/S1064827595287997.

Frederic Lauwers, Francis Cassot, Valerie Lauwers-Cances, Prasanna Puwanarajah, and Henri Duvernoy. Morphometry of the human cerebral cortex microcirculation: General characteristics and space-related profiles. *NeuroImage*, 39(3):936–948, 2008. ISSN 10538119. doi: 10.1016/j.neuroimage.2007.09.024.

A. A. Linninger, I. G. Gould, T. Marinnan, C. Y. Hsu, M. Chojecki, and A. Alaraj. Cerebral microcirculation and oxygen tension in the human secondary cortex. *Annals of Biomedical Engineering*, 41(11): 2264–2284, 2013. ISSN 00906964. doi: 10.1007/s10439-013-0828-0.

N Lyubimova and JW Hopewell. Experimental evidence to support the hypothesis that damage to vascular endothelium plays the primary role in the development of late radiation-induced cns injury. *The British journal of radiology*, 77(918):488–492, 2004.

Wayne D Newhauser and Rui Zhang. The physics of proton therapy. *Physics in Medicine & Biology*, 60 (8):R155, 2015.

WD Newhauser, C Schneider, L Wilson, S Shrestha, and W Donahue. A review of analytical models of stray radiation exposures from photon-and proton-beam radiotherapies. *Radiation protection dosimetry*, pages 1–7, 2017.

Thomas Pasko, Bradley Seidman, and Scott Birkhead. Physician characteristics and distribution in the us: 2000-2001 edition. *Chicago, IL: American Medical Association*, 2000.

Rebecca L Siegel, Kimberly D Miller, and Ahmedin Jemal. Cancer statistics, 2017. *CA: a cancer journal for clinicians*, 67(1):7–30, 2017.

FA Stewart, S Hoving, and NS Russell. Vascular damage as an underlying mechanism of cardiac and cerebral toxicity in irradiated cancer patients. *Radiation research*, 174(6b):865–869, 2010.

H A van der Vorst. *Iterative Krylov Methods for Large Linear Systems.* Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, Cambridge, UK; New York, USA, 2003. ISBN 9780521818285.

John Yarnold and Marie-Catherine Vozenin Brotons. Pathogenetic mechanisms in radiation fibrosis. *Radiotherapy and oncology*, 97(1):149–161, 2010.

A Impact and Specific Aims for Ph.D. Dissertation

Increasing patient survival rates following cancer treatment have brought more attention to the long-term effects of radiation therapy, which can detrimentally affect the patients quality of life. Radiation-induced necrosis of the brain has received increased attention in the literature due to the difficulty of diagnosis, its severe and potentially deadly impact on the patients quality of life, and concerns over its relation to proton and heavy ion therapies. Researchers have demonstrated a correlation between vascular network damage and necrosis induction. The long term goal of this project is to investigate the mechanisms linking vascular damage to radiation induced necrosis through simulation and small animal models. Radiation damage to cells in vascular tissue can cause changes to a blood vessels geometry, i.e. radius and length, and possibly lead to vessel death through rupture or occlusion. This radiation-induced vascular damage can cause local and non-local changes in blood flow through vascular remodeling and redistribution of blood flow, possibly leading to hypoxic regions in healthy tissue.

Vascular networks in an organ, such as the brain, contain billions of vessels with dimensional scales spanning four orders of magnitude. The calculation of dose to large multi-scale geometries is computationally difficult and time consuming. Recent improvements in the modeling of vascular network geometry and radiation track structure lend themselves to the investigation of multi-scaled damage in vascular networks of a human brain. However, the computational requirement such a model for the whole brain poses one of the biggest challenges. The goal of this project is to develop a computational framework for simulating vascular radiation damage, and to show that it is computationally feasible to perform these calculations for the whole brain.

To investigate the role of multi-scale vascular damage in radiation-induced necrosis, we will develop a computational framework to simulate radiation damage on whole-organ vascular networks. Using this framework, we will test the following hypothesis: It is computationally feasible to simulate multi-scale radiation-induced changes with 10% statistical uncertainty in a 3 billion vessel network, mimicking the human brain, within 3 million CPU-h using less than 4000 CPUs. We propose to the test this hypothesis by performing the following specific aims:

Specific Aim 1: Evaluate a computational figure-of-merit (composite of accuracy, speed, and efficiency) for multi-scale dose calculations of proton tracks incident on a vascular network, while limiting simulation time to 2 million CPU-h ($FOM \geq 5 \times 10^{-5}$).

Specific Aim 2: Determine if multi-scale simulations of vasculature shrinkage caused by radiation-induced DNA damage can be completed in 800,000 CPU-h for the entire brain vascular network.

Specific Aim 3: Determine if changes in blood flow through the whole-brain vascular network can be computed using steady state Poiseuille equations within 200,000 CPU-h using less than 4 TB electronic memory.

B Algorithm Descriptions

B.1 Dose Calculation Algorithm

The dose calculation algorithm is used to calculate dose to 80 points arranged around each vessel in a network. Due to the size of the targets and particle tracks, it is necessary to perform the dose calculation on a track by track basis. To optimize the dose calculation (and minimize the number of actual calculation steps) we use a recursive collision detection technique. This improve the computation of the algorithm by allowing efficient determination of where the particle track is in our geometry. Dose is computed using a simplified radiation transport algorithm. The particles have little stochastic properties and their path can be determined analytically. Additionally, the dose computation was simplified by using a symmetric radial dose model.

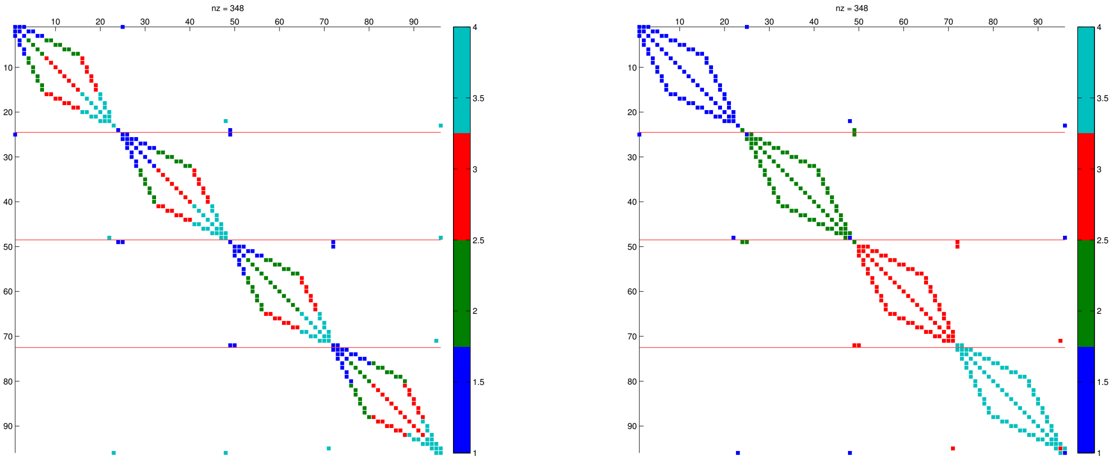
The application was optimized in two ways, first all I/O is interleaved the with the computation of dose. This is critical because of the large amounts of data that must be written to disk. Additionally, the geometric portions of the algorithm have been implemented using SIMD operations, to utilize all SIMD lanes of each core. We are however limited by use of the exponential function; which after profiling is most time-consuming part of the application.

Parallelization was performed with a hybrid approach. Each node processes multiple vessels in parallel during computation. This allows for almost a 100% core utilization during the computation portions. Work on each node is independent from other nodes. Work units are distributed to each node using small stripes from the input file. This ensures a more uniform work distribution, than using larger chunks, improving performance.

B.2 Network Partitioning Application

The first step for solving the blood flow throughout the vascular network is to reorganize the data. The input files for the algorithm require an efficient data layout to minimize the communication necessary between compute nodes during the Blood Flow Calculations. To begin, data from the input file is imported onto the compute nodes in uniform sized chunks, which results in a poor data distribution on the cluster. A k-way graph partitioning algorithm is applied to the network, grouping vessels into well-connected chunks. We used the ParMETIS libray. The data is then exported back to the file; sorted such that all vessels belonging to a particular partition are stored contiguously, enabling efficient importing into other applications.

Indexes identifying connected vessels and junctions needed to be regularized to utilize the new data layout. This is rectified in two passes. First, the indexes identifying connections between vessels were renumbered by mapping the old vessel locations in the file to the new locations. Second, the bifurcation junction indexes were renumbered. This process begins by finding any junction that is present in more than one partition. These junctions represent locations where data must be communicated when constructing the matrix. They are labeled as edge junctions and assigned a host partition where the data will be aggregated during the Blood Flow Calculations. The junctions are then re-indexed using a map constructed from a local renumbering scheme and the new edge junction identifiers. This map is applied to junction indexes of the vessels, edge junctions, and boundary condition locations.



(a) Before Preprocesssing

(b) After Preprocesssing

Figure 2: Figure a shows the node of origin for different matrix elements in the original vessel data distribution. Figure b shows element origins after the preprocessing application is used. The different colors represent the different compute nodes, in this example there are four. The horizontal red lines represent the block of rows stored on different processors, with node 1 storing the data for the top of the matrix.

B.3 Blood Flow Simulation Application

The blood flow through a vessels is modeled using the Poiseuille equation. This equation for a vessel of radius r and a length L is

$$Q = \frac{\pi r^4}{8\eta L}\left(P_{\text{in}} - P_{\text{out}}\right) \tag{2}$$

where η is the blood viscosity and P_{in} and P_{out} are the pressures at the input and output of the vessel respectively (Linninger et al., 2013).

Calculating the blood flows in the preprocessed network is broken into three parts: matrix construction, matrix inversion or pressure solving, and flow calculation. First, each compute node imports in the relevant junction indexes, edge junction indexes, boundary conditions, and vessel radii and lengths for its assigned partition. Many matrix solver packages require that each compute node contain all the information for a contiguous set of rows in the matrix, stored in compressed sparse row format. This is the motivation for preprocessing the vessel network and assigning host partitions to the edge junctions. The source of the data needed for each matrix element before and after Network Preprocessing is illustrated in Figure 2, highlighting the significantly reduced communication required to construct the matrix.

The matrix is solved using an iterative Krylov method (van der Vorst, 2003) to calculate the blood pressures at each junction.Following the matrix solving step, each compute node receives the pressure for each of its non-local edge junctions. A small matrix is constructed to calculate the volumetric flow rate through each vessel using Equation 2, enabling the use of efficient matrix-vector multiplication routines to calculate the flows in each vessel.