Request for Research Allocation on QB2:
Modeling the Blood Flow in the Whole Human Body

William P. Donahue (user:wdonah2)[1] and Wayne D. Newhauser (user: newhaus)[1,2]

[1]Department of Physics and Astronomy, Louisiana State University, Baton Rouge, LA
[2]Mary Bird Perkins Cancer Center, Baton Rouge, LA

1 Problem Statement

We are requesting 2,000,000 service units (SUs) for the final optimization and timing measurements of a scalable algorithm for simulating the blood flow through each vessel in the human body body on the LONI QB2 cluster. Blood flow is critical to supporting the function of all tissues in mammals. The human body contains approximately 20 billion vessels (Herman, 2007) with dimension from centimeters to micrometers (Gray and Lewis, 1918; Linninger et al., 2013; Lauwers et al., 2008). The blood flow can be modulated by disease processes (Ingvar and Franzén, 1974), internal and external environmental conditions (Haddy and Scott, 1968; Hauge et al., 1983), and stress and injury (Marion et al., 1991; Fassbender et al., 2001). In the United States, cardiovascular and cerebrovascular diseases resulted in approximately 775,000 deaths during 2015 (Murphy et al., 2017) and are expected to cost \$1 trillion per year by 2030 (Heidenreich et al., 2011).

The goal of this work is to simulate the biologic effect of radiation damage on vasculature, specifically the changes in blood flow. While changes in blood flow have been calculated extenively in the literature (Linninger et al., 2013; Quarteroni et al., 2016; Atkins, 1974; Formaggia et al., 1999), to our knowledge no one has attempted to model the blood flow through each vessel in the entire body. This proposal supports the final tuning of our matrix solver and timing measurements of our proposed algorithm. This work is 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 two most common approaches to blood flow modeling are 3D computational fluid dynamics simulations using finite element meshes and 1-d time dependent simulations (Quarteroni et al., 2016). While these techniques provide excellent representations of the physics involved, they are limited to modeling small numbers of vessels (<100,000) (Zagzoule and Marc-Vergnes, 1986; Sherwin et al., 2003; Pan et al., 2013; Blanco et al., 2014; Müller and Toro, 2014). An approach that has demonstrated the ability to scale to large numbers of vessels is steady-state Poiseuille flow (Linninger et al., 2013; Hyde et al., 2014).

Using a research allocation from HPC@LSU for SuperMIC (hpc_wd_vdmf01), we have demonstrated the ability to construct a simple vascular geometry that is scalable to 17 billion vessels. Using the same allocation, we have performed preliminary studies of the scalability of our blood flow model, which will be discussed later. Our algorithm consists of two applications, a network preprocessing application and a blood flow simulation application.

The resources provided by SuperMIC limited our maximum network size to 8 billion vessels, or about the number in the human brain (Lauwers et al., 2008). To take this project to the whole-body scale we need the resources of QB2, *i.e.*, access to 256 nodes.

3 Methodology

Our two applications 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 maintainted by Lawrence Livermore National Laboratory (LLNL).

4 Research Plan

To calculate the volumetric flow rates in a vascular network, a two-step algorithm is applied. The first step reorganizes the input vascular network for computational efficiency using graph partitioning algorithms. After partitioning, the data must be regularized to reflect the new data layout. The second step calculates the blood flow through the vessels. Here, the primary focus is solving a system of equations to find the pressures at each junction. These pressures are used to calculate the volumetric flow rate through each vessel. There is a separate application for each step. Timing and scalability information must be collected for both applications. A more detailed description of the algorithms is available in the Attachments.

The first thing that must be accomplished is tuning of the matrix solver. This is motivated by our preliminary scaling results discussed below. With the guidance of LLNL (the library developers), we have to complete tuning of the library options for the matrix solving. This tuning will be a systematic process focusing on getting reproducible and scalable performance from the solver. We will use small job sizes for most of the tuning runs. These will complete in less than an hour wall clock time and use less than 32 nodes. The final test will be a run with 128 nodes to confirm the application completes as expected.

Timing measurements of the solver will be performed for networks with 10 vessels to 17 billion vessels. The maximum number of nodes used will be 256. Each network size will be run 5 times, to determine the impact of system variability on the algorithm.

5 Requirements Analysis

Here we will present our preliminary results from the HPC@LSU SuperMIC cluster, to demonstrate the scalability of our algorithm.

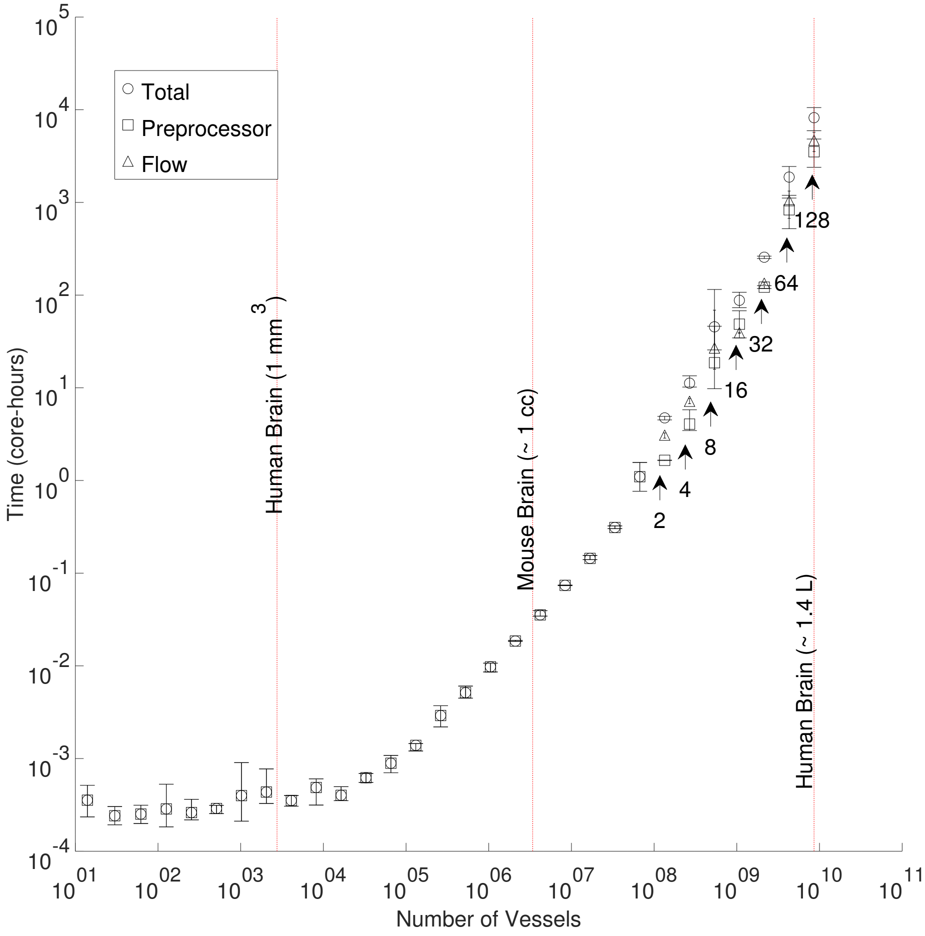


Figure 1: Plot of compute time in core-hours (T) versus the number of vessels in a network. Vertical dashed lines represent the number of vessels in various volumes. The numbers signify the quantity of compute nodes needed to perform the calculations. The error bars represent the maximum and minimum run times of the 5 timing runs.

5.1 Computational Speed

Figure 1 plots the total time to preprocess the network and perform the flow calculations. An 8 billion vessel required 4.5 wall-clock hours to complete, using 128 processors. The time of the Network Preprocessing application was dominated by I/O operations for all generations. For smaller networks, the computational speed of the Blood Flow Calculations was limited by I/O. However, the matrix solver becomes the dominant use of compute resources as the number of generations in the network increases, due to more interactions between the computer memory and the CPU. Additionally, the amount of inter-node communication required to solve the matrix increased as the problem size grew, further increasing the computation time of the solver.

5.2 Scalability

The speed up for the Network Preprocessor is plotted in Figure 2. This figure highlights that the overall scalability is limited by the I/O scalability, while all other components scaled modestly with the number of processors used. Figure 3 shows the speedup of the Blood Flow Calculations as the number of compute nodes increased. For this application, the overall scalability is not limited directly by the I/O and instead is limited by the solver for large compute node counts. The drop in scalability could be related to MPI

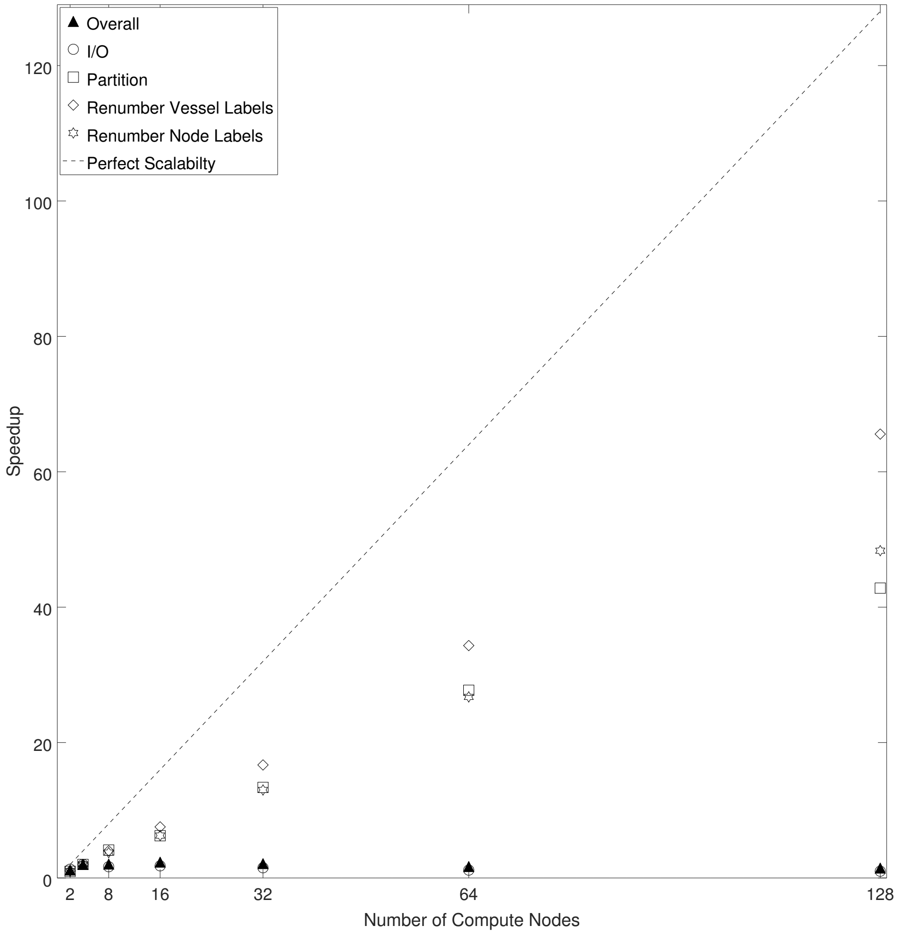


Figure 2: Plot of Speedup (S) of the Network Preprocessing application versus the number of compute nodes. the dashed line represents the ideal linear relationship related with perfect scalability. The dark triangles represent the overall scalabilit of the algorithm.

implementation details, as the library used changes some communications algorithms when the number of nodes is 64 or more. This will be investigated during the tuning process. However, we believe it is an artifact of the small workload when the problem size is broken down.

5.3 Memory Use

The Network Preprocessing application remained well below the 62 GB available on each node for computations. The Blood Flow Calculation has a peak of about 90 GB that occurs during the matrix solver setup routine. After contacting the developers of the library, the issue is a special triple vector product used in the library. This is the motivation for the new tuning campaign.

5.4 Conclusions of Requirements Analysis

We believe that, with 256 compute nodes, the algorithms listed here will be able to simulate the whole-body blood flows in 55,000 SUs. For our manuscript, we want to collect 5 timing measurements for each network size. This means we need approximately 1,000,000 SUs for our final measurements.

Due to suggestions from LLNL, tuning of the solver will be performed prior to the measurements. This will be primarily conducted on smaller network sizes requiring 2-16 nodes. Then a 128 node job will be performed to confirm the algorithm performs as expected. Based on previous experience, this will

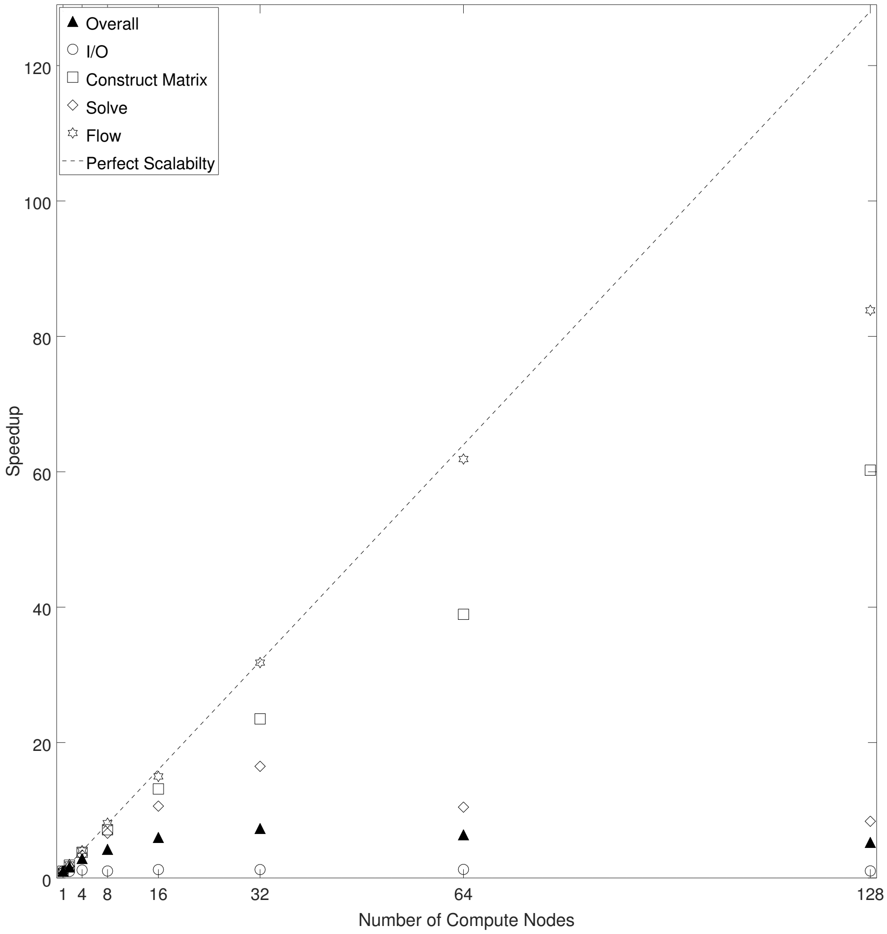


Figure 3: Plot of Speedup (S) of the Blood Flow Calculations versus the number of compute nodes. the dashed line represents the ideal linear relationship related with perfect scalability. The dark triangles represent the overall scalability of the algorithm.

require approximately 500,000 SUs. Due to the volatile nature of software development, we are requesting double this amount for the allocation.

This results in the following requested resources.

Application Tuning & Finalization	1,000,000 SUs
Timing Measurements	1,000,000 SUs
Total Requested SUs	**2,000,000 SUs**

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

G L Atkins. *Multicompartment Models for Biological Systems.* Methuen's monographs on biological subjects. Methuen, 1974. ISBN 9780412211805.

Pablo J. Blanco, Sansuke M. Watanabe, Enzo A. Dari, Marco Aurélio R F Passos, and Raúl A. Feijóo. Blood flow distribution in an anatomically detailed arterial network model: criteria and algorithms. *Biomechanics and Modeling in Mechanobiology*, 13(6):1303–1330, 2014. ISSN 16177940. doi: 10.1007/s10237-014-0574-8.

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.

K Fassbender, B Hodapp, S Rossol, T Bertsch, J Schmeck, S Schütt, M Fritzinger, P Horn, P Vajkoczy, S Kreisel, J Brunner, P Schmiedek, and M Hennerici. Inflammatory cytokines in subarachnoid haemorrhage: Association with abnormal blood flow velocities in basal cerebral arteries. *Journal of Neurology Neurosurgery and Psychiatry*, 70(4):534–537, apr 2001. ISSN 00223050. doi: 10.1136/jnnp.70.4.534.

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.

Luca Formaggia, Fabio Nobile, Alfio Quarteroni, and Alessandro Veneziani. Multiscale modelling of the circulatory system: a preliminary analysis. *Computing and Visualization in Science*, 2(2-3):75–83, dec 1999. ISSN 1432-9360. doi: 10.1007/s007910050030.

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

F J Haddy and J B Scott. Metabolically linked vasoactive chemicals in local regulation of blood flow. *Physiological reviews*, 48(4):688–707, oct 1968. ISSN 0031-9333. doi: 10.1152/physrev.1968.48.4.688.

A Hauge, G Nicolaysen, and M Thoresen. Acute effects of acetazolamide on cerebral blood flow in man. *Acta Physiol Scand*, 117(2):233–239, feb 1983. ISSN 14230313. doi: 10.1159/000139316.

Paul A Heidenreich, Justin G Trogdon, Olga A Khavjou, Javed Butler, Kathleen Dracup, Michael D Ezekowitz, Eric Andrew Finkelstein, Yuling Hong, S Claiborne Johnston, Amit Khera, Donald M Lloyd-Jones, Sue A Nelson, Graham Nichol, Diane Orenstein, Peter W.F. Wilson, and Y Joseph Woo. Forecasting the future of cardiovascular disease in the United States: A policy statement from the American Heart Association. *Circulation*, 123(8):933–944, mar 2011. ISSN 00097322. doi: 10.1161/CIR.0b013e31820a55f5.

Irving P Herman. *Physics of the human body. [electronic resource].* Biological and medical physics, biomedical engineering. Berlin ; New York : Springer, c2007., 2007. ISBN 9783540296041.

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.

Eoin R. Hyde, Andrew N. Cookson, Jack Lee, Christian Michler, Ayush Goyal, Taha Sochi, Radomir Chabiniok, Matthew Sinclair, David A. Nordsletten, Jos Spaan, Jeroen P H M Van Den Wijngaard, Maria Siebes, and Nicolas P. Smith. Multi-scale parameterisation of a myocardial perfusion model using whole-organ arterial networks. *Annals of Biomedical Engineering*, 42(4):797–811, 2014. ISSN 15739686. doi: 10.1007/s10439-013-0951-y.

D. H. Ingvar and G. Franzén. ABNORMALITIES OF CEREBRAL BLOOD FLOW DISTRIBUTION IN PATIENTS WITH CHRONIC SCHIZOPHRENIA. *Acta Psychiatrica Scandinavica*, 50(4):425–462, aug 1974. ISSN 16000447. doi: 10.1111/j.1600-0447.1974.tb09707.x.

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.

Donald W. Marion, Joseph Darby, and Howard Yonas. Acute regional cerebral blood flow changes caused by severe head injuries. *J Neurosurg*, 74(3):407–414, mar 1991. ISSN 0022-3085. doi: 10.3171/jns.1991.74.3.0407.

Lucas O Müller and Eleuterio F Toro. A global multiscale mathematical model for the human circulation with emphasis on the venous system. *International Journal for Numerical Methods in Biomedical Engineering*, 30(7):681–725, 2014. ISSN 20407939.

Sherry L Murphy, Jiaquan Xu, Kenneth D Kochanek, Sally C Curtin, and Elizabeth Arias. Deaths: Final data for 2015. Technical Report 6, National Center for Health Statistics, 2017.

Qing Pan, Ruofan Wang, Bettina Reglin, Guolong Cai, Jing Yan, Axel R. Pries, and Gangmin Ning. A One-Dimensional Mathematical Model for Studying the Pulsatile Flow in Microvascular Networks Not Exhibiting Vascular Tone. *Journal of Biomechanical Engineering*, 136(1):011009, 2013. ISSN 0148-0731. doi: 10.1115/1.4025879.

A. Quarteroni, A. Veneziani, and C. Vergara. Geometric multiscale modeling of the cardiovascular system, between theory and practice, 2016. ISSN 00457825.

S. J. Sherwin, L. Formaggia, J. Peiró, and V. Franke. Computational modelling of 1D blood flow with variable mechanical properties and its application to the simulation of wave propagation in the human arterial system. *International Journal for Numerical Methods in Fluids*, 43(6-7):673–700, oct 2003. ISSN 02712091. doi: 10.1002/fld.543.

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.

M Zagzoule and J P Marc-Vergnes. A global mathematical model of the cerebral circulation in man. *Journal of biomechanics*, 19(12):1015–1022, 1986. ISSN 0021-9290.

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 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.

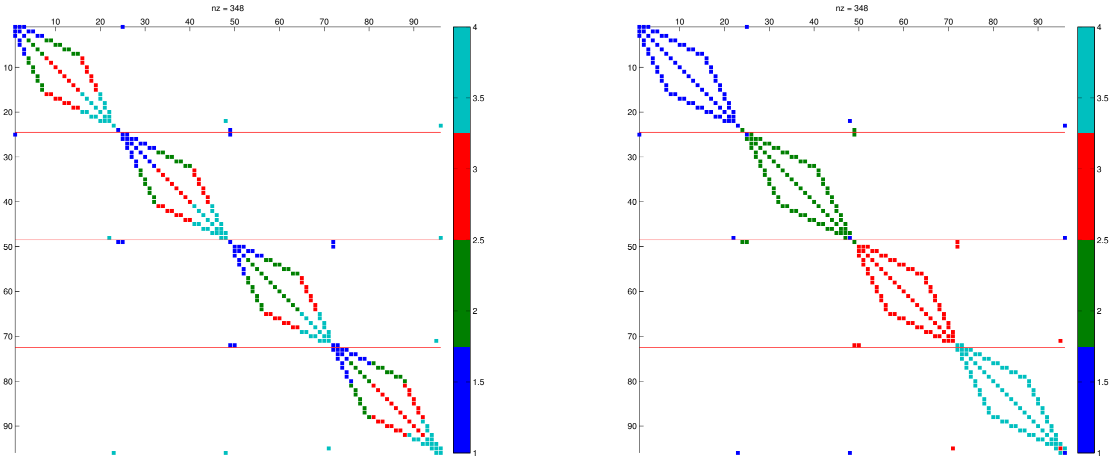
B.2 Blood Flow Simulation Application

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

$$Q = \frac{\pi r^4}{8\eta L} \left(P_{\text{in}} - P_{\text{out}}\right) \tag{1}$$

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



(a) Before Preprocesssing

(b) After Preprocesssing

Figure 4: 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.

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 4, 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 1, enabling the use of efficient matrix-vector multiplication routines to calculate the flows in each vessel.