Simulation Framework for Vascular Radiation Damage

Storage Allocation Proposal

PI: Wayne Newhauser

Abstract

We request a Storage allocation of 2000 GB for a Ph.D. Thesis Project in the Medical Physics Department. This document will outline the overview and impact of the project, the projected storage requirements, and data retrieval procedure when the allocation leaves. The project will see active development over the next two years so additional storage will be requested. This storage allocation is being requested to support the SupreMic Allocation "hpc_wd_vdmf01."

Project Description

Increasing patient survival rates following cancer treatment have brought more attention to the long-term effects of radiation therapy, which can detrimentally affect the patient's 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 patient's 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 vessel's 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. Figure 1 shows a possible series of events that could connect radiation damage in vascular cells to radiation-induced necrosis.

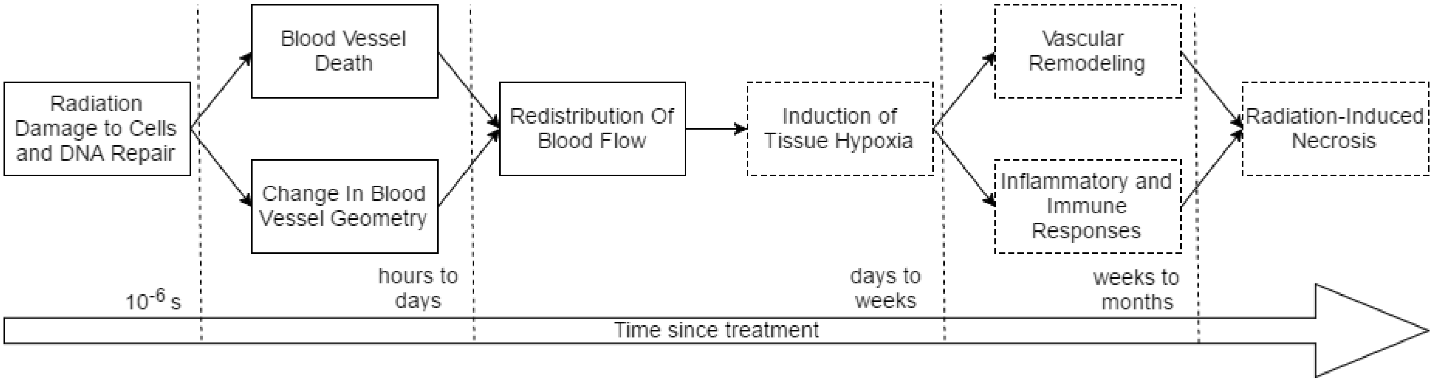
Radiation Damage to Cells and DNA Repair
Blood Vessel Death
Change In Blood Vessel Geometry
Redistribution Of Blood Flow
Induction of Tissue Hypoxia
Vascular Remodeling
Inflammatory and Immune Responses
Radiation-Induced Necrosis
10^{-6} s
hours to days
days to weeks
weeks to months
Time since treatment

Figure 1: A diagram of some of the possible connections between radiation damage to cells and vascular tissues and radiation necrosis. The dashed boxes represent aspects of the hypothesis that are not the focus of the work described in this proposal.

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 512 GB electronic memory.

Current Project Standing

The project is beginning its focus on Aim 3. In this aim the goal is to demonstrate that we can compute the steady-state mass flow rate through the number of vessels in a human brain. The general premise is to solve a 4 billion by 4 billion matrix and then perform a matrix vector multiplication. Even in sparse notation the matrix has approximately 15.6 billion non-zero elements (336 GB). This means that we need to utilize a distributed matrix scheme to solve the system. In its current implementation the code uses HDF5 (The HDF Group. 2016) for data storage and I/O operations. These operations are performed in parallel using collective I/O. The matrix is constructed in a node by node manor that is currently serial in nature. The matrix equations are solved using the Intel MKL Direct Sparse Solver for Clusters algorithm. The latest beta version of the Intel MKL library must be used to enable solving of the matrix without over running over the Mother-Superior Node Resources. Using this code, simulations have been performed at the small scale on a personal desktop computer.

Allocation Description and Use

This storage allocation will be used to store the libraries, compilers, source code, and data for a Ph.D. project entitled “Multi-scale Simulation of Vascular Damage Following Irradiation.” The expanded storage is required to enable the student to install the components required for linking and compiling of the project code in a system independent way. Below is a table outlining the components to be stored on the system and their approximate sizes.

Item to be Installed	Size on Disk (GB)
Compilers	
Intel C++ Compiler 2018 update 1 beta	11
Libraries	
Intel MKL Library Cluster Edition – 2018 update 1 beta	1.8
Intel MPI – 2018 update 1 beta	0.5

Intel Performance Primitives – 2018 update 1 beta	3.0
HDF5 version 1.10	0.094
HYPRE (Distributed Sparse Solver)	0.5
Google Test Framework for C++	0.02
Glibc-static (possibly GCC as well)	4 (max)
Tools	
Cmake version 3.7	0.06
Project Source Code	2
Project Data (currently -may need more in the future)	1700
Total	1800

Based on the calculations above the request for 2000 GB is a conservative estimate of the current needs of the project. Additional libraries can be installed in the allocation if needed for other portions of the project. The project data estimate is for the current portion of the project, this might need to be expanded once additional components of the project become available and online.

The primary motivation for the large Project data space is to reduce the need to reconstruct base data for each trial run. The codes used to generate this data for our largest dataset size take approximately 5 hours to run. This generates an 850 GB HDF5 file. It takes approximately 1 hour to copy such a file, saving a lot of time during the other jobs being run, i.e. not having 128 nodes waiting for a file copy at the beginning of a job. As the goal of the jobs is to collect multiple runtime measurements for scalability studies, the file will be copied 5 times during the application run saving up to 25 hours of computation time.

Number of Users

The Ph.D. student will be the only user of this storage allocation. His contact and identification information can be found below.

William Donahue
wdonah2@lsu.edu
Phone: (860) 690-1566
HPC username: wdonah2

Data Handling

During project

All libraries, tools, and compilers will be installed to the partition using the /work directory as a scratch space. Transfer of the source code will be handled with rsync to reduce communication and storage requirements. Project data will be backed up to the project directory after completing generation. Files will be copied to the work directory when jobs are run to reduce waiting of system resources.

After Project

All libraries installed during the time of the project will be removed. Additionally, all source code will be backed up and any necessary project data will be downloaded using ftp or other transfer protocol to storage drives on a local machine or an external backup drive.