CATEGORY: SUPERCOMPUTING - SC09
POSTER P5284
CONTACT NAME Xuan Huang: hu6@umbc.edu

GPU TECHNOLOGY CONFERENCE

UMBC
AN HONORS UNIVERSITY IN MARYLAND

Long-time Simulation of Advection-Diffusion-Reaction System using FEM and FVM on Hybrid CPU/GPU Nodes

Xuan Huang and Matthias K. Gobbert, Department of Mathematics and Statistics, University of Maryland, Baltimore County
High Performance Computing Facility, www.umbc.edu/hpcf

CIRC

Background

Calcium induced calcium release (CICR) in a heart cell can be modeled by a system of coupled time-dependent advection-diffusion-reaction equations:

$$u_t^{(i)} - \nabla \cdot \left(D^{(i)}\, \nabla u^{(i)}\right) + \beta^{(i)} \cdot \left(\nabla u^{(i)}\right) = q^{(i)}$$

for the concentrations $u^{(i)}(\mathbf{x}, t)$, $i = 1, \ldots, n_s$, of the n_s reactive species
for all points $\mathbf{x} \in \Omega \subset \mathbb{R}^d$ $(d = 1, 2, 3)$ and time $0 < t \leq t_{\text{fin}}$.
The right hand sides $q^{(i)}$, $i = 1, \ldots, n_s$, are given by

$$q^{(i)} = r^{(i)}(u^{(1)}, \ldots, u^{(n_s)}) + s^{(i)}(u^{(i)}, \mathbf{x}, t) + f^{(i)}(\mathbf{x}, t).$$

- diffusivity matrices $D^{(i)} \in \mathbb{R}^{d \times d}$ diagonal positive definite
- constant advection vector $\boldsymbol{\beta}^{(i)} \in \mathbb{R}^d$,
- linear source term $f^{(i)}(\mathbf{x}, t)$ for test problems
- non-linear reaction terms $r^{(i)}$ couples all PDEs
- application term $s^{(1)}(u^{(1)}, \mathbf{x}, t)$ associated with calcium, $s^{(i)} \equiv 0$ for $i > 1$
- Key term JSR in $s^{(1)}$ models superposition of CRU injection at all $\hat{\mathbf{x}} \in \Omega_s$ as point sources:

$$J_{\text{SR}}(u^{(1)}, \mathbf{x}, t) = \sum_{\hat{\mathbf{x}} \in \Omega_s} g\, S_{\hat{\mathbf{x}}}(u^{(1)}, t)\, \delta(\mathbf{x} - \hat{\mathbf{x}})$$

Numerical Methods

Two method of line approaches:

- Spatial discritization using Finite Elements, take advantage of the regular shape of the domain Ω and use a uniform mesh of 3-D brick elements.
- Using Finite Volume method, integrate equations over each control volume, using numerical flux functions to approximate diffusive and advective fluxes.

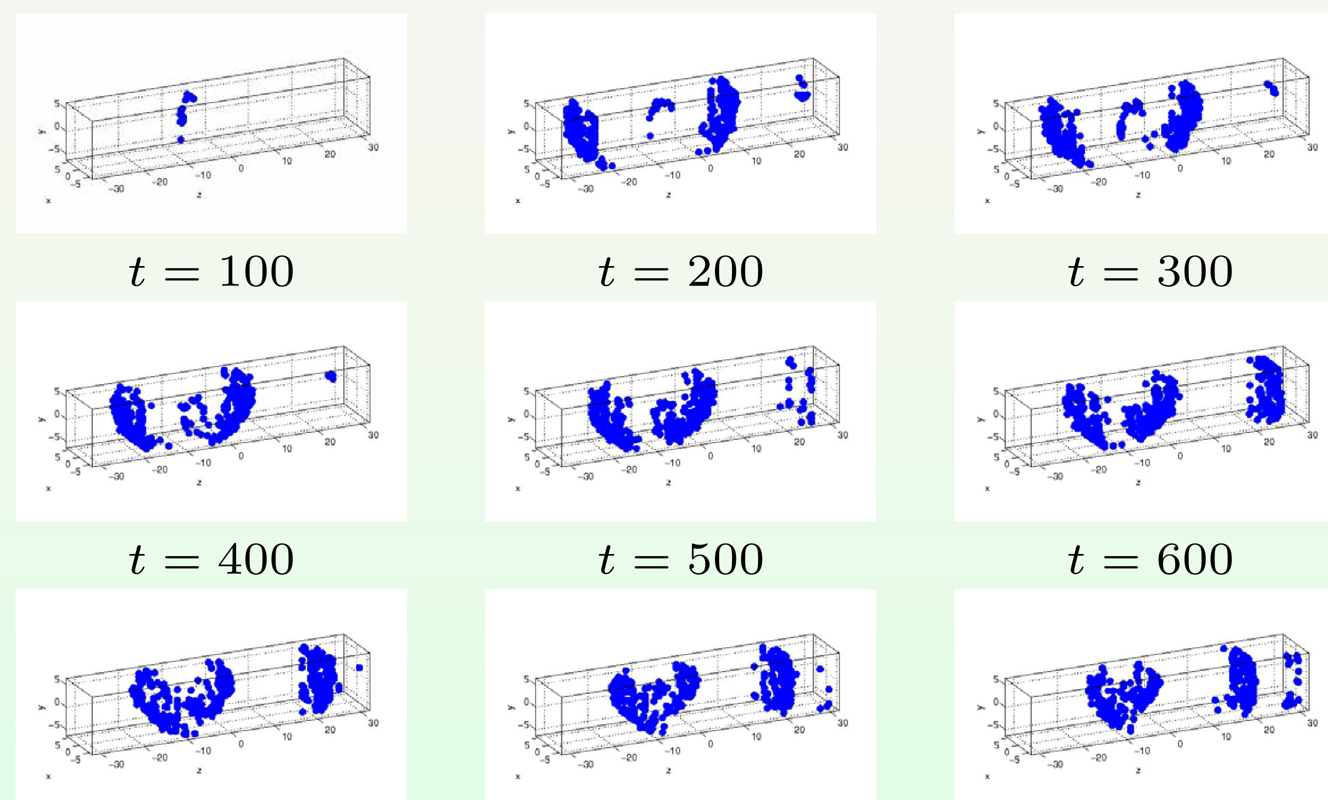
Summary of other numerical methods:

- Time stepping for a large system of stiff ODEs, fully implicit with automatic step size (and order for NDFk) selection: $1 \leq k \leq 5$,
- Non-linear solver: Newton method with analytical Jacobian
- Linear solver: iterative Krylov subspace method family (specifically BiCGSTAB) with matrix-free products for all system matrices and their transposes

$N_x \times N_y \times N_z$	DOF	number of time steps	memory usage predicted (GB)
$32 \times 32 \times 128$	421,443	58,416	0.05
$64 \times 64 \times 256$	3,257,475	73,123	0.41
$128 \times 128 \times 512$	25,610,499	89,088	3.24

Simulation of CICR in 3D

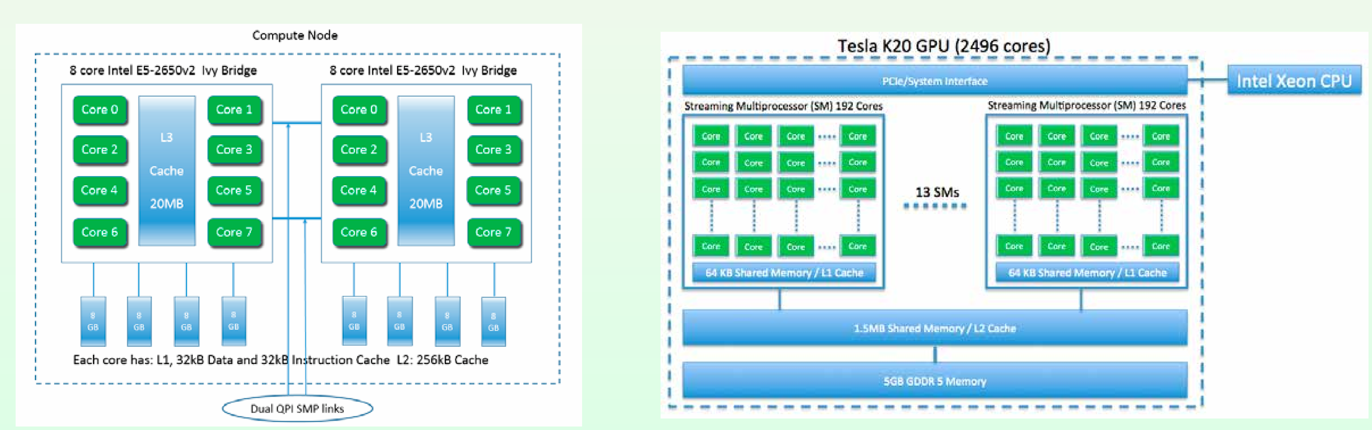
Open calcium release units throughout the cell using finite element method. Based on simulation with CUDA + MPI.



$t = 100$ $t = 200$ $t = 300$
$t = 400$ $t = 500$ $t = 600$
$t = 700$ $t = 800$ $t = 900$

Hybrid Node Schematic

- Each node contains two eight-core Intel E5-2650v2 Ivy Bridge CPUs. The 64 GB of the node's memory are connected to CPUs via 8 memory channels.
- The two CPUs of a node are connected to each other by two QPI (quick path interconnect) links.
- 19 hybrid nodes, each contain two NVIDIA K20 GPUs with 2496 computational cores and 5 GB of global memory.
- Nodes connected via quad-data rate (QDR) InfiniBand interconnect.
- Storage of more than 750 TB connected by IB.



References

- A memory-endcient finite volume method for advection-diffusion-reaction systems with non-smooth sources. Numer. Methods Partial Differential Equations. Volume 31, Issue 1, pages 143–167, January 2015
- Technical Report HPCF–2014–8

CUDA + MPI

The program used to perform the parallel computations presented here is an extension of a C program with MPI for parallel communications. However, to enable efficient calculations on GPU, almost all calculations have been redesigned to take advantage of GPU parallelism.

- Each node has 2 eight-core CPUs, each CPU is connected to a NVIDIA K20 GPU via PCIe bus. Therefore, the program is designed to have 2 processes on each node, each process utilize the full power of a unique GPU.
- Minimize data transfer between host and device:
  - Allocate all large arrays on device memory, avoid data transfer in loops
  - Share C struct between host and device
  - Only communicate for output and data exchange between processes
- Reduce time required for MPI communication:
  - Use Non-blocking MPI communication such as `MPI_Isend` and `MPI_Irecv`, and start MPI command as early as possible
  - Keep GPUs busy while performing MPI communication
  - Call `MPI_Waitall` right before computations involving MPI communicated data

Results

Wall clock time for FEM on hybrid CPU/GPU in HH:MM:SS and speedup against CPU only runtime on one 16-core node.

nodes (GPU/node)	$32 \times 32 \times 129$	$64 \times 64 \times 256$	$128 \times 128 \times 512$
1 node (16 cores)	00:20:28	02:36:56	42:07:19
1 node (1 GPU)	00:42:33 (0.48)	01:58:19 (1.33)	25:09:06 (1.67)
1 node (2 GPUs)	00:43:42 (0.47)	01:25:32 (1.83)	13:46:41 (3.06)
2 node (1 GPU)	00:36:24 (0.56)	01:13:29 (2.13)	13:25:47 (3.14)
2 node (2 GPUs)	00:40:21 (0.50)	01:02:11 (2.52)	07:47:28 (5.41)
4 node (1 GPU)	00:29:20 (0.69)	00:49:29 (3.17)	07:19:21 (5.75)
4 node (2 GPUs)	00:39:35 (0.51)	00:51:35 (3.04)	04:41:47 (8.97)

Wall clock time for FVM on hybrid CPU/GPU in HH:MM:SS and speedup against CPU only runtime on one 16-core node.

nodes (GPU/node)	$32 \times 32 \times 129$	$64 \times 64 \times 256$	$128 \times 128 \times 512$
1 node (16 cores)	00:12:04	02:24:00	28:13:32
1 node (1 GPU)	00:31:09 (0.39)	02:00:32 (1.19)	16:22:51 (1.72)
1 node (2 GPUs)	00:30:33 (0.39)	01:30:03 (1.60)	08:58:16 (3.15)
2 node (1 GPU)	00:23:46 (0.51)	01:15:14 (1.91)	08:39:43 (3.26)
2 node (2 GPUs)	00:28:08 (0.42)	01:05:04 (2.21)	05:03:27 (5.58)
4 node (1 GPU)	00:20:46 (0.58)	00:52:09 (2.76)	04:45:01 (5.94)
4 node (2 GPUs)	00:28:02 (0.43)	00:55:10 (2.61)	03:05:28 (9.13)

Acknowledgments

- HPCF www.umbc.edu/hpcf
- REU Site www.umbc.edu/hpcreu
- CIRC www.umbc.edu/circ
- NSF (MRI, SCREMS), NSA, UMBC