Data Management Challenges in HBPS

Jong Youl Choi¹, Michael Churchill², Davide Currel³, Sonata Mae Valaitis⁴, Robert Hager⁵, Seung-Hoe Ku⁶, E. D’Azevedo⁷, Bill Hoffman⁸, David Pugmire⁹, Scott Klasky⁵, C. S. Chang³

¹ORNL, ²PPPL, ³Univ. of Illinois Urbana-Champaign, ⁴Kitware

XGC I/O Performance

We maintain cutting edge I/O performance for XGC on various file systems, including SSDs and NVMe, on Cori, Theta, and Summit.

The Fusion HPBS project is focusing on researching multi-way coupling science to study multi-scale/multi-physics.

1) XGC and hPIC
- Plasma-material-interaction hPIC code coupled into XGC
- hPIC code has 6D marker particles, while XGC has 5D marker particles

2) XGC and F-analysis coupling
In XGC and F analysis coupling, we move the F computation to a dedicated analysis code. XGC asynchronously offloads those computations via ADIOS and improves computational performance.

Our team continues to innovate to take full advantage of the new memory and storage technologies, and to provide the highest levels of performance.

XGC Software Process

Agile XGC development
- Incorporate a modern CMake build system
- Continuous Integration testing system
- Git workflow incorporated with CI system
- Integrate CDash into github

Coupling Workflows

The Fusion HPBS project is focusing on researching multi-way coupling science to study multi-scale/multi-physics.

1) XGC and hPIC
- Plasma-material-interaction hPIC code coupled into XGC
- hPIC code has 6D marker particles, while XGC has 5D marker particles

2) XGC and F-analysis coupling
In XGC and F analysis coupling, we move the F computation to a dedicated analysis code. XGC asynchronously offloads those computations via ADIOS and improves computational performance.

EFFIS

EFFIS is an integrated platform of services to compose, launch, monitor, and control coupled applications. EFFIS can simplify the complexity of composing, running, and monitoring applications on HPC systems. We integrate HBPS with EFFIS to “easily” compose coupled HBPS workflows on HPC Resources (Cori, Theta, and Summit).

EFFIS’s using a python-like interface can allow “easy” integration to visualization tools (Visit, Python notebooks).

Research Details

a) To improve movement performance and flexibility, HBPS integrated with ADIOS for data management.
b) Developing multi-way coupling science cases to study multi-scale/multi-physics scenarios.
c) Exploiting data locality to improve performance
- XGC computes 5D f and electromagnetic field
- Hand-off computational reduction of physics from XGC
- Analysis code consumes in-memory f data

Acknowledgments: Work supported by U.S. DOE Office of Science, ASCR and FES. This research used resources of OLCF, ALCF, and NERSC, which are DOE Office of Science User Facilities.
Performance Enhancements of XGC

E. D’Azevedo¹, A. Scheinberg², M. Shephard³, P. Worley⁴, S. Sreepathi⁵, B. MacKie-Mason⁵, T. Williams⁵, and the SciDAC HBPS XGC Team


Funding is from DOE ASCR and FES Offices

XGC Meshing

- Improved mesh quality in areas where flux curves interact with reactor wall
- Improved matched mesh gradation at x-point
- Reordering of mesh data for better memory access during XGC simulations

Improved mesh gradation at X-point

XGC based on Parallel Unstructured Mesh PIC (PUMPiPic)

PUMPiPic - Components to support PIC operations on distributed unstructured meshes (2D and 3D)
- Mesh centric – no independent particle structure
- Distributed mesh with overlaps (PICparts)
- Particle migration and load balancing between pushes
- Adjacency-based particle containment determination
- Focused on structures for execution on GPUs
- Omega GPU ready mesh topology being integrated
- Particles stored by element in new SCS data structure
- Test shows on-par performance using less memory

<table>
<thead>
<tr>
<th>no sorting</th>
<th>full sorting</th>
</tr>
</thead>
<tbody>
<tr>
<td>pttcs (Ki)</td>
<td>time (s)</td>
</tr>
<tr>
<td>128</td>
<td>2.298661</td>
</tr>
<tr>
<td>256</td>
<td>2.895464</td>
</tr>
<tr>
<td>512</td>
<td>3.79263</td>
</tr>
<tr>
<td>1024</td>
<td>4.972283</td>
</tr>
<tr>
<td>2048</td>
<td>7.089673</td>
</tr>
<tr>
<td>4096</td>
<td>11.57884</td>
</tr>
</tbody>
</table>

Implementing XGC physics and Numerics with PUMPiPic:
- Since all core data structures are changed code, code being rewritten in C++

Status of implementation:
- Based on original PUMI structures - new GPU focused structures will be integrated when complete
- Core mesh/particle interaction operations in place
- Mesh evolve in place
- Ion and electron push (including subcycling) implemented
- Initial if simulations executed
- Performance evaluation and improvement underway
- Initial push results show 25% improvement on many core system
- Other steps slower due to need to modify mesh copies (underway)

Snapshot of electrostatic potential fluctuation (a) at toroidal angle ζ<sub>0</sub>,θ<sub>0</sub>,φ<sub>0</sub> from left to right and (b) in local domain of each group at ζ<sub>0</sub>

XGC on Summit

- XGC is part of Early Science Programs on Summit, Aurora and Perlmutter
- XGC is an ECP code
- XGC uses an unstructured grid in poloidal plane, each MPI rank gets particles from a section of poloidal plane
- Main computational kernel is electron push
- Utilizes Kokkos

XGC core/push_e.F90:

```fortran
subroutine push_e
  call sort_particles ; Sort particles by grid cell
  do i=1, n_particles ; Loop over particles
    do j=1, n_cycles ; Subcycle electrons
      do k=1, n_runge_kutta ; RK4 loop
        isearch(b,l) ; determine which grid cell particle
        call gather_field ; Interpolate field at particle location
        call calculate_dx ; Solve physics: dt/dx = f(x,..)
        call advance_particles ; Update particle position and velocity
      end do
    end do
  end do
end subroutine push_e
```

Good Weak Scaling to Full Summit

- On 256 nodes of Summit, GPU version has 15X speedup over CPU only
- Good weak scaling up to full Summit using 1.24 trillion electrons on GPU and 1.24 trillion ions on CPU

Performance on KNL

- Kokkos version of XGC has been ported to Cori KNL
- Roofline analysis of vectorized version of XGC shows in-lining and re-factoring useful in optimizing use of wide-vector registers. However, vector dependences and data type conversions limiting peak performance

Performance Analytics for Computational Experiments for XGC

- Central hub of performance data, already used in Climate application
- Interactively deep-dive and track performance benchmark
- Facilitate performance analysis:
  - Load balancing
  - Identification of bottlenecks
  - Inform targeted optimization efforts

Details on XGC-Kokkos

- XGC in Fortran, Kokkos in C++
- Fortran interface (Cabana) enables easy porting of new kernels
- Single code for CPU and GPU
- Electron push kernel in CUDA Fortran (C++ version under development)

Performance on Summit+

- Good weak scaling using less memory
- Predefined Fortran type for vector dependences and data type conversions limiting peak performance

Performance on KNL

- Kokkos version of XGC has been ported to Cori KNL
- Roofline analysis of vectorized version of XGC shows in-lining and re-factoring useful in optimizing use of wide-vector registers. However, vector dependences and data type conversions limiting peak performance

Performance Analytics for Computational Experiments for XGC

- Central hub of performance data, already used in Climate application
- Interactively deep-dive and track performance benchmark
- Facilitate performance analysis:
  - Load balancing
  - Identification of bottlenecks
  - Inform targeted optimization efforts

https://pace.ornl.gov

Tree and Flame Graphs