DESIGN AND DEVELOPMENT OF THE THERMOPHYSICS UNIVERSAL RESEARCH FRAMEWORK

Robert Martin¹, J. Tran¹, J. Arakai¹, H. Le¹, D. Bilyeu²
PI: Jean-Luc Cambier²

ERC Inc.¹, In-Space Propulsion Branch²,
Air Force Research Laboratory
Edwards Air Force Base, CA USA

RQ-West 6.1 Research Review,
Institute of Pure and Applied Math,
University of California, Los Angeles
January 20, 2014

AFTC/PA clearance No. 15014, 16 January 2015
Spacecraft Propulsion Relevant Plasma:
- From hall thrusters to plumes and fluxes on components
- Complex reaction physics i.e. Discharge and Breakdown in FRC
- Relevant Densities often Span 6+ Orders of Magnitude
- Spatial scales of interest span $\mu m$-100m range

Electric Propulsion Plumes

FRC

Chamber Environment
Spacecraft Propulsion Relevant Plasma:

- From hall thrusters to plumes and fluxes on components
- Complex reaction physics i.e. Discharge and Breakdown in FRC
- Relevant Densities often Span 6+ Orders of Magnitude
- Spatial scales of interest span $\mu m$-100m range

Solution?
Adaptive Multi-Scale/Physics Algorithms
Spacecraft Propulsion Relevant Plasma:
- From hall thrusters to plumes and fluxes on components
- Complex reaction physics i.e. Discharge and Breakdown in FRC
- Relevant Densities often Span 6+ Orders of Magnitude
- Spatial scales of interest span $\mu m$-100 m range

Solution?
Adaptive Multi-Scale/Physics Algorithms

But which Ones?
Research Framework Motivations

- Optimal Algorithm not known a Priori
- Multiple Research Codes from Group
- Redundant Functionality
- Complexity of GPU Support
Research Framework Motivations

- Optimal Algorithm not known a Priori
- Multiple Research Codes from Group
- Redundant Functionality
- Complexity of GPU Support
- Needed Non-ITAR Subset for Collaborators: TURF
- Minimal Framework to Capture Algorithms for SM/MURF
Evolution of Computing Hardware:

Early Coliseum Development – 2001

11/2001: ASCI White - #1 Supercomputer (Top500.org)
- 7.2 TFlop/s, 8192 Cores
- 512-Nodes x 16-Cores

Courtesy of Lawrence Livermore National Laboratory, U.S. Dept. of Energy
Evolution of Computing Hardware:

Early Framework Development – 2012

11/2012: Titan - #1 Supercomputer (Top500.org)
17,590 TFlop/s, 560,640 Cores
18,688-Nodes x (16-CPU-Cores+14-GPU-K20x-SMs)

Courtesy of Oak Ridge National Laboratory, U.S. Dept. of Energy
Why not upgrade current software?

Evolution of Computing Hardware:

Early Framework Development – 2012

11/2012: Titan - #1 Supercomputer (Top500.org)
17,590 TFlop/s, 560,640 Cores
18,688-Nodes x (16-CPU-Cores+14-GPU-K20x-SMs)

Courtesy of Oak Ridge National Laboratory, U.S. Dept. of Energy
Evolution of Computing Hardware:

Early Framework Development – 2012

11/2009: Jaguar - #3 Supercomputer (Top500.org)

1,759.0 TFlop/s, 224,256 Cores

18,688-Nodes x 12-CPU-Cores

Courtesy of Oak Ridge National Laboratory, U.S. Dept. of Energy
Why not Upgrade Current Software?

Evolution of Computing Hardware:
Early Framework Development – 2012
11/2012: Titan - #1 Supercomputer (Top500.org)
17,590 TFlop/s, 560,640 Cores
18,688-Nodes x (16-CPU-Cores+14-GPU-K20x-SMs)

Courtesy of Oak Ridge National Laboratory, U.S. Dept. of Energy
Homogeneous: Coliseum

Architecture: MPI+CPU

Heterogeneous: TURF

Architecture: MPI+CPU+GPU
ARCHITECTURE → FRAMEWORK

Homogeneous: Coliseum

Framework: MPI+CPU

Heterogeneous: TURF

Framework: MPI+CPU+GPU
Homogeneous: Coliseum

Framework:
MPI+CPU+???

Heterogeneous: TURF

Framework:
MPI+CPU+GPU+Load Balance
Code Object Structure:

- Similar Structures across XFluid/CudaPIC/Fluidsolver Codes
- Natural Hierarchical Tree-Structure
- All Domain Member Objects
  Similar Structure: Info+Data Matrices
Example: Particle Distribution Object

Data Object Shared Common Structure:

- **Info Data:**
  - Intrinsic Properties
  - Data Size Information

- **Data Blocks:**
  - kMatrix Arrays of Bulk Data
  - Often Share Common Sizes
    (i.e. Number of Particles/Field Size/etc.)

PDFist Object

(Info)

- Np - Number of Particles
- Z - Particle Charge
- m - Particle Mass

kMatrixND

- Pos[Np] - Particle Positions
- Mom[Np] - Particle Momentums
- W[Np] - Computational Weights
**Example: kMatrix Object**

kMatrix Follows Same Structure:

- kMatrix Adds Copy On/Off GPU
- Similar Functionality Partially Implemented in PDist/PSort/Field/etc.
- Data Objects All Need this Functionality
- Functionality Reproduced Repeatedly Across Objects

---

**kMatrix Object**

(Info)

- \(Z\) - Total Number of Elements
- \(N0:N(d)\) - Number of Elements/dim
- \(S0:S(d)\) - First Element/dim
- \(E0:E(d)\) - Last Element/dim

**kMatrixND**

- \(hA[Z]\) - Host Data Array
- \(dA[Z]\) - Device Data Array

**Utility Functions**

- MakeDeviceCopy()
- CopyDeviceToHost()
- CopyHostToDevice()
- FreeDeviceCopy()
- Size()
**Example: kMatrix Object**

kMatrix Follows Same Structure:

- kMatrix Adds Copy On/Off GPU
- Similar Functionality Partially Implemented in PDist/PSort/Field/etc.
- Data Objects All Need this Functionality
- Functionality Reproduced Repeatedly Across Objects

**kMatrix Object**

- **Info**
  - \( Z \) - Total Number of Elements
  - \( N0:N(d) \) - Number of Elements/dim
  - \( S0:S(d) \) - First Element/dim
  - \( E0:E(d) \) - Last Element/dim

- **kMatrixND**
  - \( hA[Z] \) - Host Data Array
  - \( dA[Z] \) - Device Data Array

- **Utility Functions**
  - MakeDeviceCopy()
  - CopyDeviceToHost()
  - CopyHostToDevice()
  - FreeDeviceCopy()
  - Size()
Formalized Service Objects:

- Unify Natural Tree Hierarchy
- Provides Standard GPU Load/Unload Functionality
- Addition of Standard Name/Type-Name Enables High Level Access
- Enables Object Addressing from Root Similar to AMR Domain Functionality in Research Code
Encapsulated TURF Data Structures:

- Objects from Original Structure Inherit from GS-Object
- Enables Common Reuse of Utility Interface
- Also Enables Recursive Copy and Structure Traversal
CPU/GPU POLYMORPHISM

Why “Core()”?

- Most Kernels Equivalent to Nested For Loops
- “Core()” Enables “Apply()” GPU and CPU Routes to Same Code
- GPU Code Requires No Data Dependency
- GPU Routine OK on CPU
- Even Non-Ideal Provides Baseline Performance
Generic Operator → Stacks

- Common “Apply()” Enables Execution of Generic Operators
- Generic Operators may be Stacked
- Stacks Executed in Stages between Communication
- Multi-Domain/Process → All Domains to Barrier and Sync
- Mixed Hyperbolic/Parabolic Solves Require Flexible Communications
Text to Objects:

- Operations Generically Parsed into Parameter Maps
- Parameter Maps fed to OpFactory
- OpFactory Clones Operators by ’TYPE’ String
- Operators Initialized via Parameter Map
- Operators Placed in Stack for Current Stage & Domain

Same Executable Runs:
Euler, Vlasov, DSMC, or ES-PIC
... Only Change is Input File
Increased Complexity Second Attempt...

- Spherical Potential: $\Phi \propto 1/r$
Increased Complexity Second Attempt...

- Spherical Potential: $\Phi \propto 1/r$
- Slight $\Delta X \rightarrow$ Orbits Unsynchronized
**Spherical Test Case**

Increased Complexity Second Attempt...

- Spherical Potential: $\Phi \propto \frac{1}{r}$
- Slight $\Delta X \rightarrow$ Orbits Unsynchronized

Without Merge & Split
Increased Complexity Second Attempt...

- Spherical Potential: $\Phi \propto 1/r$
- Slight $\Delta X \rightarrow$ Orbits Unsynchronized
- Periodic with Full Nonlinear-C.N. Push

Without Merge & Split
Increased Complexity Second Attempt...

- Spherical Potential: \( \Phi \propto \frac{1}{r} \)
- Slight \( \Delta X \) → Orbits Unsynchronized
- Periodic with Full Nonlinear-C.N. Push
- Adding Merge & Split:
  - Energy Conservation \textbf{OK}
  - Accuracy: Scatter + \textbf{Dispersion}?
Spherical Test Case

Increased Complexity Second Attempt...

- Spherical Potential: $\Phi \propto \frac{1}{r}$
- Slight $\Delta X \rightarrow$ Orbits Unsynchronized
- Periodic with Full Nonlinear-C.N. Push
- Adding Merge & Split:
  - Energy Conservation OK
  - Accuracy: Scatter + Dispersion?

 Including Merge & Split

Source of Dispersion?
Examples of TURF Results:

- ES-PIC Comparison to ICEPIC
- DSMC Shock on Satellite
- Collisionless Shock PIC
- Collisionless Shock Vlasov
- Collisionless Shock Comparison
Thank You

Questions?