Center For Compressible Multiphase Turbulence

S. Balachandar and H. Lam
Department of Mechanical and Aerospace Engineering
Department of Electrical and Computer Engineering
University of Florida
Demonstration Problem

- Integrated simulations
- Experimental measurements for validation
We desire to perform predictive simulation of these flows with as much multi-scale physics as possible.
Multiscale Coupling Strategy

Atomistic
Quantum and MD

EOS, Thermodynamic and transport properties, shock Hugoniot

Continuum Scale Modeling and Simulations

Macroscale
> O(10^9) particles
Macro LES of turbulence
Point-particle approximation

Microscale
O(1) – O(10^4) particles
Fully resolved, DNS

Mesoscale
O(10^5) – O(10^8) particles
Well resolved interface turbulence
Unresolved particulate turbulence (Meso-LES)

Particle-flow mass, momentum and energy coupling models

Multiphase LES closure models for interface & particulate turbulence

CCMT
How Different Pieces Fit

Exascale Emulation

Uncertainty Budget

Integrated Code
Integrated Simulations

Proxy-apps
Energy-efficient algorithms

Load balance, etc.

Computer Science

CMT Physics & Experiments

Simulation Roadmap

UQ: Uncertainty Quantification
UR: Uncertainty Reduction
DB: Dakota Bundles
CS: Concurrent Simulations
Cylindrical Explosive Dispersal of Particles up to 200μs

Features:
- 30 Millions computational cells
- 5 Millions computational particles
- $r_{\text{max}} = 0.30\text{cm}$
Cylindrical Explosive Dispersal of Particles up to 1ms

Features:
- 2.5 Millions computational cells in a (r,θ) plane
- 1 Millions computational particles
- $r_{\text{max}} = 0.60\text{cm}$
Uncertainty Quantification of Explosive Dispersal

Later time, looking only at The gas phase

A small perturbation in the charge has no influence on the particle dispersal
Uncertainty Reduction – Microscale Exp/Sim

- **Physics – Informed Hybrid Surrogate Modeling**

\[
F(t) = 6\pi \mu a \left(\overline{u^S} - u^P\right) + V \left(\rho \frac{Du}{Dt} V\right) + V \int_{\xi=-\infty}^{t} \left(K_{IU} \left(\frac{t - \xi}{a}\right) \frac{c_0}{a} \right) \left(\frac{d\rho u}{dt} \right) \frac{c_0}{a} d\xi \\
+ \int_{\xi=-\infty}^{t} \left(K_{IU} \left(\frac{t - \xi}{a}\right) \frac{c_0}{a} \right) \left(\frac{d^2 \rho x}{dt^2} \right) \frac{c_0}{a} d\xi
\]

Data:
Sun et al. (2005)
\[
M_s = 1.22 \\
d = 8 \text{ mm}
\]
Multi-Particle (FCC) / Shock Interaction
Multi-Particle Force Analysis

- Inviscid flow with a shock Mach number of 1.22
- 10% volume fraction (FCC arrangement)

Force histories
Decision Making with Uncertainty Budget

- Decision making based on uncertainty of prediction metrics
  - Uncertainty Reduction
  - Validation & Uncertainty Quantification
  - Uncertainty Reduction
    - Macroscale Experiments/Simulations
    - Validation UQ & Uncertainty Propagation
    - Macroscale Experiments/Simulations
    - Validation UQ & Uncertainty Propagation
    - Microscale Experiments/Simulations

Experiment Roadmap

Simulation Roadmap
Validation and UQ Framework

Sandia Multiphase Shock Tube (Wagner et al. 2011)

- Experiments
- Numerical Simulation
- Surrogate Model
- Measurement
- Uncertainty in Input
- Measured Input
- Measurement Uncertainty in Input
- Measured Prediction
- Measurement Uncertainty in Prediction

- Prediction Metrics
- Physical Model Error
- Numerical Model Error
- Numerical Solution Error
- Propagated Uncertainty

- % of total uncertainty
  - Downstream
  - Upstream

- Time (sec)

- Uncertainty in input
- Uncertainty in measurement
Update

- **First-year accomplishments:**
  - Fully staffed by April 2014
  - Executed large simulations on several DOE Platforms
  - Dakota bundled runs for UQ
  - UB framework and demonstration for shock-curtain interaction problem
  - Improved physics-based hybrid surrogate models from micro and mesoscale simulations

- **Going forward**
  - Hero runs of demonstration problem with extreme resolution and additional physics
  - Vigorous uncertainty reduction through micro/meso modeling
  - Emulation-informed CMT-Nek development for near-future architectures
CS Research @ CCMT

- Hardware-software co-design of CMT applications (performance, energy, and thermal issues)

- Behavioral emulation for design-space exploration of CMT applications
Goal & Research Thrusts

Goal: Perform multi-objective optimization of CMT apps to provide inputs

- For Exascale CMT app development
- For NGEE Behavioral Emulation: co-design, design-space exploration

Research thrusts

- Multi-objective optimization
  - Energy, performance
  - Based on thermal constraints
- Code generation for hybrid multicores
  - Support for multiple types of cores
  - Support for vectorization
- Load balancing
CMT-NEK - Autotuning

Characteristics of CMT-Nek: Nested loops

Loop Transformation: CHiLL
- Unroll & Permutations

Genetic Algorithm: Rapid search to optimum

3D Matrix multiplication kernel

Genetic Algorithms

Loop transformations

Code generator

Transformed matrix multiplication code

Search Engine

Empirical Performance Evaluation

Optimized matrix multiplication code

Best performing version
Results For Derivative Operation

- Up to 45% improvement in energy and performance
- Platform: AMD Fusion @ Sandia National Laboratory
- Similar improvement observed on other platforms such as IBM BG/Q and AMD Opteron.
Update: CS Research for CMT-Nek

• **First-year accomplishments:**
  – *Autotuning framework* based on genetic algorithms
    • 3D matrix multiplication for derivatives and interpolation

• **Going forward**
  – *Autotune* other key computational and communicational kernels of CMT-Nek
  – *Investigate* performance, energy and power models for hybrid multicores
  – *Investigate* thermal models and constraints
**CS Research @ CCMT**

- Hardware-software co-design of CMT applications (performance, energy, and thermal issues)

- Behavioral emulation for design-space exploration of CMT applications
Context: DOE Co-design

(One of) the difficulties of co-design

Co-design gets more difficult the further you get from open collaboration and the closer you get to the “truth” (particularly with competing vendors and national security applications)

- ASC concerns
- Vendor concerns

- National Security Applications
- Unclassified, but not open applications
- Open Co-design
  - Released Proxy Apps
  - Open vendor information
- Standard NDA
- Deep NDA

- ASC: Involve staff with clearances in co-design efforts and developing proxy apps
- Vendor: Limit number of lab staff engaging in multiple “deep NDA” discussions

CCMT Behavioral Emulation Flow

Application Design-space Exploration

- Apps & Kernels
  - skeleton apps (macro-scale)
  - mini-apps (meso-scale)
  - kernels (micro-scale)

Architecture Design-space Exploration

- Notional systems & Notional Architectures

Future-gen Systems & Notional Architectures

Application BEOs*
AppBEOs

Architecture BEOs*
ArchBEOs

Simulation/Emulation Platform

Behavioral simulation (SW) or emulation (HW) experimentation

Testbed benchmarking & experimentation

* BEO – Behavioral Emulation Object
**Behavioral Emulation (BE)**

- **Component-based, coarse-grained simulation**
  - Fundamental constructs called BE Objects (BEOs) act as surrogates
  - BEOs characterize & represent behavior of app, device, node, & system objects as fabrics of interconnected ArchBEOs (with AppBEOs) up to Exascale

- **Multi-scale simulation**
  - Hierarchical method based upon experimentation, abstraction, exploration

- **Multi-objective simulation**
  - Performance, power, reliability, and other environmental factors

<table>
<thead>
<tr>
<th>Level</th>
<th>Applications</th>
<th>Architecture</th>
<th>BEO Models</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Macro Level</strong></td>
<td>Skeleton-apps</td>
<td><strong>System</strong> BEO fabrics</td>
<td>- Models abstracted from Meso-scale&lt;br&gt;- Testbed experimentation in support&lt;br&gt;- Notional <em>Exascale system</em> exploration</td>
</tr>
<tr>
<td><strong>Meso Level</strong></td>
<td>Mini-apps</td>
<td><strong>Node</strong> BEO fabrics</td>
<td>- Models abstracted from Micro-scale&lt;br&gt;- Testbed experimentation in support&lt;br&gt;- Notional <em>Exascale node</em> exploration</td>
</tr>
<tr>
<td><strong>Micro Level</strong></td>
<td>Kernels</td>
<td><strong>Device</strong> BEO fabrics</td>
<td>- Architectural studies&lt;br&gt;- Testbed experimentation as foundation&lt;br&gt;- Notional <em>Exascale device</em> exploration</td>
</tr>
</tbody>
</table>
Research Thrusts

1. Behavioral Characterization
   – How do we build, calibrate, then validate performance?

2. Parameter Estimation
   – How do we efficiently capture behavior in surrogates?

3. Synchronization & Congestion
   – How do we handle sync and congestion at scale?

4. Resilience & Energy
   – How do we extend beyond performance attributes?

5. Management & Visualization
   – How do we measure & analyze massive systems & apps?

6. Reconfigurable Architectures
   – How do we exploit FPGA hardware for speed & scale?
Behavioral Emulation Workflow (to date)

Step 1: Calibration
- BEOs: computation & communication
- Performance models
  1. Sample on target platforms for interpolation
  2. Use Kriging method for multi-dim interpolation
  3. Evaluate & recalibrate, if necessary

Step 2: Validation
- Microbenchmarks
  - Computation
  - Communication
- Kernels
  - 2D matrix multiply
  - Sobel filtering
  - 3D matrix multiply
- Platforms
  - Tile-Gx36
  - Xeon Phi

Step 3: Prediction
- Kernels on
  - Next-gen Tile-Gx72
- Kernels on
  - Anticipated Intel Knight’s Landing

1. Sample on target platforms for interpolation
2. Use Kriging method for multi-dim interpolation
3. Evaluate & recalibrate, if necessary
Behavioral Emulation Platforms

- **Software PDES** behavioral simulator
  - **Initial prototype**: In-house developed SMP simulator
  - **V2**: Leverage existing PDES simulators (e.g., SST, ROSS)

- **Hardware-accelerated** behavioral emulator
  - **FPGA-based** reconfigurable computing
  - Leverage emerging reconfigurable supercomputing advances (e.g., UF’s Novo-G, Microsoft’s Catapult)
## Software PDES Simulator: Sample Results

### Tile 6x6 (existing sys.)

<table>
<thead>
<tr>
<th>Operation</th>
<th>Simulated Time (ns)</th>
<th>Prediction Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D MM 1024x1024</td>
<td>2.82x10^9</td>
<td>-0.35%</td>
</tr>
<tr>
<td>Sobel 800x600</td>
<td>9.27x10^7</td>
<td>-2.61%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Simulation Time</th>
<th>SMP Simulation Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPGA</td>
<td></td>
</tr>
<tr>
<td>35.7us</td>
<td>4.82ms</td>
</tr>
<tr>
<td>~135x</td>
<td></td>
</tr>
<tr>
<td>54.2us</td>
<td>8.08ms</td>
</tr>
<tr>
<td>~149x</td>
<td></td>
</tr>
</tbody>
</table>

### Tile 9x8 (next-gen)

<table>
<thead>
<tr>
<th>Operation</th>
<th>Simulated Time (ns)</th>
<th>Prediction Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D MM 1024x1024</td>
<td>1.66x10^9</td>
<td>To be determined</td>
</tr>
<tr>
<td>Sobel 800x600</td>
<td>2.58x10^7</td>
<td>To be determined</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Simulation Time</th>
<th>SMP Simulation Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPGA</td>
<td></td>
</tr>
<tr>
<td>81.1us</td>
<td>10.24ms</td>
</tr>
<tr>
<td>~126x</td>
<td></td>
</tr>
<tr>
<td>102.0us</td>
<td>17.94ms</td>
</tr>
<tr>
<td>~176x</td>
<td></td>
</tr>
</tbody>
</table>

### KNL 9x8 (anticipated)

<table>
<thead>
<tr>
<th>Operation</th>
<th>Simulated Time (ns)</th>
<th>Prediction Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D MM 1024x1024</td>
<td>5.87x10^8</td>
<td>To be determined</td>
</tr>
<tr>
<td>Sobel 800x600</td>
<td>1.37x10^7</td>
<td>To be determined</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Simulation Time</th>
<th>SMP Simulation Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPGA</td>
<td></td>
</tr>
<tr>
<td>81.1us</td>
<td>10.24ms</td>
</tr>
<tr>
<td>~126x</td>
<td></td>
</tr>
<tr>
<td>102.0us</td>
<td>17.94ms</td>
</tr>
<tr>
<td>~176x</td>
<td></td>
</tr>
</tbody>
</table>
### Single-FPGA Results: 3 Data Points

<table>
<thead>
<tr>
<th>Tile 6x6 (existing sys)</th>
<th>Simulated Time</th>
<th>Prediction Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D MM 1024x1024 across 36 cores</td>
<td>2.82x10^9 ns</td>
<td>-0.35%</td>
</tr>
<tr>
<td>Sobel 800x600 image across 36 cores</td>
<td>9.27x10^7 ns</td>
<td>-2.61%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FPGA Simulation Time</th>
<th>SMP Simulation Time</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>35.7 us</td>
<td>4.82 ms</td>
<td>~135x</td>
</tr>
<tr>
<td>54.2 us</td>
<td>8.08 ms</td>
<td>~149x</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tile 9x8</th>
<th>Simulated Time</th>
<th>Prediction Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D MM 1024x1024 across 72 cores</td>
<td>1.66x10^9 ns</td>
<td>To be determined</td>
</tr>
<tr>
<td>Sobel 800x600 image across 72 cores</td>
<td>2.58x10^7 ns</td>
<td>To be determined</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FPGA Simulation Time</th>
<th>SMP Simulation Time</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>81.1 us</td>
<td>10.24 ms</td>
<td>~126x</td>
</tr>
<tr>
<td>102.0 us</td>
<td>17.94 ms</td>
<td>~176x</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KNL 9x8</th>
<th>Simulated Time</th>
<th>Prediction Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D MM 1024x1024 across 72 cores</td>
<td>5.87x10^8 ns</td>
<td>To be determined</td>
</tr>
<tr>
<td>Sobel 800x600 image across 72 cores</td>
<td>1.37x10^7 ns</td>
<td>To be determined</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FPGA Simulation Time</th>
<th>SMP Simulation Time</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>81.1 us</td>
<td>10.24 ms</td>
<td>~126x</td>
</tr>
<tr>
<td>102.0 us</td>
<td>17.94 ms</td>
<td>~176x</td>
</tr>
</tbody>
</table>
**Update: BE Modeling Research**

- **First-year accomplishments:**
  - Demonstrated successful *device-level* calibration, validation, & prediction
  - On existing (Xeon Phi, Tilera) & notional devices

- **Going forward**
  - Extend methodology beyond device level *(node, system)*
  - Abstraction, scalable synchronization and congestion issues

- **CMT-centric**
  - CMT kernel, proxy apps *(Nekbone)*
  - **Questions to be answered** for application design-space exploration ("knobs" for tunable design parameters)
Update: Platform Research

- **First-year accomplishments:**
  - **Software PDES** behavioral simulator
    - Proof of concept prototype: in-house developed SMP simulator
  - **Hardware-accelerated** behavioral emulator
    - Single-FPGA prototype: feasibility study with promising results

- **Going forward**
  - **Software PDES** behavioral simulator
    - V2: Leverage existing PDES simulators (e.g., SST, ROSS)
  - **Hardware-accelerated** behavioral emulator
    - Extend to multiple FPGAs on Novo-G#
    - Leverage emerging reconfigurable supercomputing advances (e.g., IBM’s CAPI coherent accelerator interface, Microsoft’s Catapult, Micron’s HMC)
Questions?