TST Meeting
November 5-6, 2015
Agenda for TST Site Visit November 5 and 6, 2015

Thursday November 5, 2015

7:45    Van pick up at University Hilton
8:00-9:00 Full Breakfast (site visit team caucus with AST members)
9:00-9:40 CCMT Overview/Highlights
9:40-10:30 Macroscale Team
10:30-10:45 Coffee break
10:45-11:45 Microscale Team
11:45-1:00 Lunch
1:00-1:30 ASU Experiments
1:30-2:00 CMT-nek
2:00-2:50 Exascale – Part 1
2:50-3:00 Coffee break
3:00-3:50 Exascale – Part 2
3:50-4:00 Short Break
4:00-4:30 Internship presentations by Chris Hajas, Chris Neal, Carlo Pascoe and Nalini Kumar
4:30-5:00 Marianne Francois (LANL) and Mark Schraad (LANL)
5:15    Transportation to University Hilton
6:00-7:30 Dinner at the University Hilton, hosted by CCMT Faculty
7:30    TST members transported back to hotel
Friday November 6, 2015

7:45    Van pickup at University Hilton
8:00-8:30  Continental Breakfast
8:30-9:45  UB Team
9:45-10:30  CS
10:30-11:00  Discussions between TST and CCMT PIs

11:00    Box Lunch; Transportation as need to hotel and/or airport
Attendee List for TST Meeting November 5-6, 2015

**Faculty**

<table>
<thead>
<tr>
<th>Name</th>
<th>Institution</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>S. Balachandar “Bala”</td>
<td>University of Florida</td>
<td><a href="mailto:bala1s@ufl.edu">bala1s@ufl.edu</a></td>
</tr>
<tr>
<td>Rafi Hafika</td>
<td>University of Florida</td>
<td><a href="mailto:haftka@ufl.edu">haftka@ufl.edu</a></td>
</tr>
<tr>
<td>Nam-Ho Kim</td>
<td>University of Florida</td>
<td><a href="mailto:nkim@ufl.edu">nkim@ufl.edu</a></td>
</tr>
<tr>
<td>Herman Lam</td>
<td>University of Florida</td>
<td><a href="mailto:hlam@ufl.edu">hlam@ufl.edu</a></td>
</tr>
<tr>
<td>Sanjay Ranka</td>
<td>University of Florida</td>
<td><a href="mailto:ranka@cise.ufl.edu">ranka@cise.ufl.edu</a></td>
</tr>
<tr>
<td>Bertrand Rollin</td>
<td>Embry-Riddle</td>
<td><a href="mailto:bertrandrollin@gmail.com">bertrandrollin@gmail.com</a></td>
</tr>
<tr>
<td>Greg Stitt</td>
<td>University of Florida</td>
<td><a href="mailto:gstitt@ece.ufl.edu">gstitt@ece.ufl.edu</a></td>
</tr>
<tr>
<td>Tom Jackson</td>
<td>University of Florida</td>
<td><a href="mailto:tlj@ufl.edu">tlj@ufl.edu</a></td>
</tr>
<tr>
<td>Siddharth Thakur “ST”</td>
<td>University of Florida</td>
<td><a href="mailto:sst@ufl.edu">sst@ufl.edu</a></td>
</tr>
</tbody>
</table>

**TST Members**

<table>
<thead>
<tr>
<th>Name</th>
<th>Institution</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>David Daniel</td>
<td>LANL</td>
<td><a href="mailto:ddd@lanl.gov">ddd@lanl.gov</a></td>
</tr>
<tr>
<td>Sam Schofield</td>
<td>LLNL</td>
<td><a href="mailto:schofield5@llnl.gov">schofield5@llnl.gov</a></td>
</tr>
<tr>
<td>Mark Schraad</td>
<td>LANL</td>
<td><a href="mailto:schraad@lanl.gov">schraad@lanl.gov</a></td>
</tr>
<tr>
<td>Justin Wagner</td>
<td>Sandia</td>
<td><a href="mailto:jwagner@sandia.gov">jwagner@sandia.gov</a></td>
</tr>
<tr>
<td>Greg Weirs</td>
<td>Sandia</td>
<td><a href="mailto:vgweirs@sandia.gov">vgweirs@sandia.gov</a></td>
</tr>
</tbody>
</table>

**AST Members**

<table>
<thead>
<tr>
<th>Name</th>
<th>Institution</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jim Costa</td>
<td>Sandia</td>
<td><a href="mailto:jecosta@sandia.gov">jecosta@sandia.gov</a></td>
</tr>
<tr>
<td>Nels Hoffman</td>
<td>LANL</td>
<td><a href="mailto:nmh@lanl.gov">nmh@lanl.gov</a></td>
</tr>
<tr>
<td>Dan Nikkel</td>
<td>LLNL</td>
<td><a href="mailto:nikkel1@llnl.gov">nikkel1@llnl.gov</a></td>
</tr>
</tbody>
</table>

**NNSA Visitors**

<table>
<thead>
<tr>
<th>Name</th>
<th>Institution</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marianne Francois</td>
<td>LANL</td>
<td><a href="mailto:mmfran@lanl.gov">mmfran@lanl.gov</a></td>
</tr>
<tr>
<td>Kevin Ruggirello</td>
<td>Sandia</td>
<td><a href="mailto:kruggir@sandia.gov">kruggir@sandia.gov</a></td>
</tr>
<tr>
<td>Kambiz Salari</td>
<td>LLNL</td>
<td><a href="mailto:salari1@llnl.gov">salari1@llnl.gov</a></td>
</tr>
<tr>
<td>Shane Schumacher</td>
<td>Sandia</td>
<td><a href="mailto:scschum@sandia.gov">scschum@sandia.gov</a></td>
</tr>
</tbody>
</table>

**Research Staff**

<table>
<thead>
<tr>
<th>Name</th>
<th>Institution</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subramanian Annamalai</td>
<td>University of Florida</td>
<td><a href="mailto:subbu.ase@gmail.com">subbu.ase@gmail.com</a></td>
</tr>
<tr>
<td>Tania Banerjee</td>
<td>University of Florida</td>
<td><a href="mailto:tmishra@cise.ufl.edu">tmishra@cise.ufl.edu</a></td>
</tr>
<tr>
<td>Charles Cook</td>
<td>University of Florida</td>
<td><a href="mailto:crcook@ufl.edu">crcook@ufl.edu</a></td>
</tr>
<tr>
<td>Jason Hackl</td>
<td>University of Florida</td>
<td><a href="mailto:jason.hackl@ufl.edu">jason.hackl@ufl.edu</a></td>
</tr>
<tr>
<td>Chanyoung Park</td>
<td>University of Florida</td>
<td><a href="mailto:cy.park@ufl.edu">cy.park@ufl.edu</a></td>
</tr>
<tr>
<td>Mrugesh Shringarpure</td>
<td>University of Florida</td>
<td><a href="mailto:mrugeshs@ufl.edu">mrugeshs@ufl.edu</a></td>
</tr>
</tbody>
</table>
Students

Georges Akiki  University of Florida  georges.h.akiki@gmail.com
Kasim Alli  University of Florida  kalli@ufl.edu
Saptarshi Biswas  University of Florida  sappy91@ufl.edu
Paul Crittenden  University of Florida  pcritte@ufl.edu
Angela Diggs  University of Florida  angela.diggs.1@us.af.mil
Brad Durant  University of Florida  neoncrash@ufl.edu
Giselle Fernandez  University of Florida  gisellefernandez@ufl.edu
Mohamed Gadou  University of Florida  mgadou@ufl.edu
Christopher Hajas  University of Florida  chrishajas@ufl.edu
Rahul Koneru  University of Florida  rahul.koneru@ufl.edu
Nalini Kumar  University of Florida  nkumar@hcs.ufl.edu
Tsing-Hsin Kuo  University of Florida  tkuo001@ufl.edu
Goran Marjanovic  University of Florida  gmarjanovic@ufl.edu
Yash Mehta  University of Florida  ymeheta@ufl.edu
Christopher Neal  University of Florida  chrislea@gmail.com
Sam Nili  University of Florida  samaunnili@ufl.edu
Frederick Ouellet  University of Florida  f.ouellet@ufl.edu
Carlo Pascoe  University of Florida  poppyc@ufl.edu
Na Qiu  Tong-Ji University (China)  nqiu@ufl.edu
Dylan Rudolph  University of Florida  rudolph@hcs.ufl.edu
Parth Shah  University of Florida  parthdshah@ufl.edu
Prashanth Sridharan  University of Florida  shan1130@ufl.edu
Cameron Stewart  University of Florida  csstewart10@ufl.edu
Yiming Zhang  University of Florida  yiming90521@gmail.com
Heather Zunino  Arizona State University  heatherzunino@gmail.com
David Zwick  University of Florida  dpzwick@ufl.edu

Administration Staff

Hollie Starr  University of Florida  hstarr@ufl.edu
CCMT
Overview and Integration

S. Balachandar

AST Meeting Agenda

Thursday
- Overview and Integration
- Macroscale
- Microscale
- Lunch
- ASU Experiments
- CMT-nek
- Exascale
- Internship (4) presentations
- DOE laboratory talks
- Dinner at University Hilton with CCMT Faculty and Staff

Friday
- Uncertainty Budget
- CS
- Discussions between TST and CCMT PIs
- Lunch and transportation to airport
Outline

- Demonstration problem
- Sequence of events and physics models
- Simulation roadmap
- Uncertainty quantification and reduction
- Y1.5 accomplishments
- Goals for next year
Demonstration Problem

- Explosive-driven cylindrical annulus of particles

- Integrated effort toward predictive simulations
- Experimental measurements for validation
Sequence of Events

- **Detonation phase**
  - Hot, dense, high pr gas
  - Shock wave

- **Compaction/collision phase**
  - Metal particles
  - Explosive material

- **Dispersion phase**
  - Shock wave

Multiscale Problem

- **Atomistic Scale**
  - Cluster of particles $O(1) - O(10^3)$

- **Microscale**
  - Instabilities $O(10^5) - O(10^7)$

- **Mesoscale**
  - Experimental setup $O(10^9)$

- **Overall Flow**
  - Explosive
  - Fluid
  - Particles

- **Length**
  - $10^0 m$ to $10^3 m$

- **Time**
  - $10^{-3} s$ to $10^{-1} s$
**Multiscale Integration Strategy**

- **Atomistic**
  - Quantum and MD

- **Continuum Scale Modeling and Simulations**
  - EOS, Thermodynamic and transport properties, shock Hugoniot

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

- **Mesoscopic**
  - $O(10^3) - O(10^6)$ particles
  - Well resolved interface turbulence
  - Unresolved particulate turbulence (Meso-LES)

- **Microscopic**
  - $O(1) - O(10^7)$ particles
  - Fully resolved, DNS

- **Particle-flow mass, momentum and energy coupling models**

---

**Physical Models – Sources of Error**

- **T1**: Detonation model
- **T2**: Multiphase turbulence model
- **T3**: Thermodynamic & transport model
- **T4**: Collision model
- **T5**: Compaction model
- **T6**: Point particle force model
- **T7**: Point particle heat transfer model

- **Detonation phase**
  - Explosive material
  - Hot, dense, high pr gas

- **Compaction/collision phase**
  - Shock wave
  - Metal particles

- **Dispersion phase**
  - T8: Deformation model
Sources of Errors & Uncertainties

- T1: Detonation modeling
- T2: Multiphase turbulence modeling
- T3: Thermodynamics & transport properties
- T4: Particle-particle collision modeling
- T5: Compaction modeling (dense-to-dilute transition)
- T6: Point-particle force modeling
- T7: Point-particle thermal modeling
- T8: Particle deformation and other complex physics
- T9: Discretization and numerical approximation errors
- T10: Experimental and measurement errors & uncertainties

Advance state-of-the-art in multiphase turbulence and point-particle models

Uncertainty Budget – Overall Plan

- Integrates all the center activities
- Uncertainty reduction through iterative improvement
Multiscale Uncertainty Propagation

Calibration

Model development

Characterization

Microscale

Mesoscale

Macroscale

**Multiscale Uncertainty Propagation**

**Calibration**

T2: Multiphase turbulence model calibration

T4: Particle collision model calibration

T3: Thermodynamics and transport properties

**Model development**

T1: Detonation model sensitivity analysis

T5: Compaction model

T8: Particle deformation and fragmentation model

**Characterization**

Microscale

Mesoscale

Macroscale

**Simulation Roadmap**

**Year 1**

**Capabilities**

- Lumped detonation
- Euler
- AUSM
- Ideal gas
- Unsteady forces
- Simple collision
- Super particles

**Demonstration Simulations**

- Hero Runs (1)
  - Grid: 30M, 30M
  - Cores: O(10K)
  - Bundled Runs (40)
    - Grid: 5M, 1M
    - Cores: O(1K)

**Experiment**

- Eglin, ASU
- SNL

**Micro/Meso Simulations**

- Shock/contact over regular array
  - Single deformable particle
- Shock curtain interaction

**Urban**

T1, T3, T9, T10

T1, T3, T4, T9

T2, T4, T6, T9

T2, T5, T8, T9

T2, T6, T7, T10

**Year 2**

**Capabilities**

- Program Run
- Navier Stokes
- AUSM
- Improved forces
- Improved collision
- Extended particles

**Demonstration Simulations**

- Hero Runs (3)
  - Grid: 100M, 30M
  - Cores: O(50K)
  - Bundled Runs (60)
    - Grid: 25M, 25M
    - Cores: O(50K)

**Experiment**

- Eglin, ASU
- SNL

**Micro/Meso Simulations**

- Shock/contact over random
  - Few deformable particles
- Instabilities of rapid dispersion

**Year 3**

**Capabilities**

- Program Run
- Multiphase LES
- AUSM
- Improved flux
- Real gas
- Improved forces
- Granular theory
- Lagrangian remap

**Demonstration Simulations**

- Hero Runs (5)
  - Grid: 150M, 100M
  - Cores: O(100K)
  - Bundled Runs (80)
    - Grid: 50M, 25M
    - Cores: O(100K)

**Experiment**

- Eglin, ASU
- SNL

**Micro/Meso Simulations**

- Turbulence over random cluster
  - Deformable random cluster
- Fan curtain interaction

**Year 4**

**Capabilities**

- Program Run
- Improved LES
- Improved flux
- Multi-component
- Stochastic forces
- DEM collision
- Lagrangian remap
- Dense-to-dilute

**Demonstration Simulations**

- Hero Runs (3)
  - Grid: 300M, 200M
  - Cores: O(300K)
  - Bundled Runs (60)
    - Grid: 100M, 70M
    - Cores: O(300K)

**Experiment**

- Eglin, ASU
- SNL

**Micro/Meso Simulations**

- Turbulence over moving cluster
  - Under-expanded multiphase jet
  - Onset of RT/RM turbulence

**Year 5**

**Capabilities**

- Program Run
- Improved LES
- Improved flux
- Multi-component
- Stochastic forces
- DEM collision
- Lagrangian remap
- True geometry

**Demonstration Simulations**

- Hero Runs (3)
  - Grid: 50M, 50M
  - Cores: O(100K)
  - Bundled Runs (100)
    - Grid: 150M, 150M
    - Cores: O(1M)

**Experiment**

- Eglin, ASU
- SNL

**Micro/Meso Simulations**

- Turbul/shock over moving cluster
  - Multiphase detonation
  - RT/RM multi-phase turbulence
Uncertainty Budget drives yearly simulation

- T1 – T10 will be computed Year-2 to Year-5

Simulation Roadmap

- Year 1 Capabilities
  - Lumped detonation
  - Euler
  - AUSM
  - Ideal gas
  - Unsteady forces
  - Simple collision
  - Super particles
  - Hero Runs (1)
    - Grid: 30M, 5M
    - Bundled Runs (30)
    - Grid: 5M, 1M
    - Cores: O(1K)

- Year 2 Capabilities
  - Program burn
  - Navier Stokes
  - AUSM+up
  - Real gas
  - Improved forces
  - Improved collision
  - Extended particles
  - Hero Runs (3)
    - Grid: 100M, 30M
    - Bundled Runs (50)
    - Grid: 25M, 10M
    - Cores: O(50K)

- Year 3 Capabilities
  - Program burn
  - Multiphase LES
  - AUSM+up
  - Real gas
  - Improved forces
  - Improved collision
  - Extended particles
  - Hero Runs (3)
    - Grid: 150M, 100M
    - Bundled Runs (60)
    - Grid: 50M, 25M
    - Cores: O(100K)

- Year 4 Capabilities
  - Stochastic burn
  - Multiphase LES
  - AUSM+up
  - Real gas
  - Stochastic forces
  - Improved flux
  - Improved collision
  - Lagrangian remap
  - Dense-to-dilute
  - Hero Runs (5)
    - Grid: 300M, 200M
    - Bundled Runs (60)
    - Grid: 100M, 70M
    - Cores: O(300K)

- Year 5 Capabilities
  - Stochastic burn
  - Improved LES
  - Improved flux
  - Multi-component
  - Stochastic forces
  - DEM collision
  - Lagrangian-remap
  - True geometry
  - Hero Runs (5)
    - Grid: 500M, 500M
    - Bundled Runs (100)
    - Grid: 150M, 150M
    - Cores: O(1M)
## Timeline: T1-T10 Uncertainty Reduction

<table>
<thead>
<tr>
<th>Task</th>
<th>Year1</th>
<th>Year2</th>
<th>Year3</th>
<th>Year4</th>
<th>Year5</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1: Detonation Sensitivity Simulation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T2a: Expansion-Fan ASU Exp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T2b: Expansion-Fan Simulation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T3a: No-particle Explosive Exp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T3b: No-particle Explosive Sim</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T4a: Particle Curtain Sim</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T4b: Particle Curtain Sim</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T5a: Mesoscale Eglin Exp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T5b: Mesoscale Explosive Sim</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T6a: Microscale Eglin Exp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T6b: Microscale Detonation Sims</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T7: Microscale Shock Simulations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T8: Post Detonation Particle Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T9: Discretization Error Quantification</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T10: Macroscale Eglin Experiments</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T10: Macroscale Simulations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Year 1.5 Key Accomplishments

1. **Simulations**
2. **Validation Experiments**
3. **CMT-nek development**
4. **Energy & thermal aware computing**
5. **UB**
6. **Exascale emulation**
1: Demonstration Problem (Macroscale)

Goal
- Yearly perform the largest possible simulations of the demonstration problem and identify improvements to be made in predictive capability

Year 1.5
- Use existing code to perform petascale simulations of the demonstration problem
- Qualitative comparison against experimental data of Frost (PM1 & PM2)
- Develop capabilities for the next hero run: real gas EOS, AUSM+up, collision modeling

Presentation
- Subbu Annamalai

1: Mesoscale Simulations

Goal
- Perform a hierarchy of mesoscale simulations to allow rigorous validation, uncertainty quantification and propagation to the demonstration problem

Year 1.5
- Mesoscale simulations of shock propagation over a bed of particles
- Mesoscale simulations of expansion fan over a bed of particles

Presentation
- Bertrand Rollin
1: Microscale Simulations

Goals
- Perform a hierarchy of microscale simulations at conditions of relevance
- Develop extended point-particle models
- Rigorous validation, uncertainty quantification and propagation to the demonstration problem

Year 1.5
- Highly resolved simulations of shock propagation over a structured array
- Highly resolved simulations of shock propagation over a random distribution
- Binary superposition model

Presentation
- Christopher Neal
1: New Point-Particle Framework

Goals
- Perform a hierarchy of microscale simulations at conditions of relevance
- Develop extended point-particle models
- Rigorous validation, uncertainty quantification and propagation to the demonstration problem

Year 1.5
- Highly resolved simulations of shock propagation over a structured array
- Highly resolved simulations of shock propagation over a random distribution
- Binary superposition model

Presentation
- Yash Mehta

2: Validation Experiments

Goals
- Obtain validation-quality experimental measurements of the demonstration problem
- Validation-quality experiments at micro and mesoscales
- Perform shock-tube track micro- and mesoscale experiments

Year 1.5
- Several sets of experiments at Eglin AFB on micro- and mesoscale explosive dispersal experiments
- Experimental studies of gas-particle mixtures under sudden expansion at ASU

Presentation
- Heather Zunino (ASU)
Goals

- Co-design an exascale code (CMT-nek) for compressible multiphase turbulence
- Perform micro, meso and demonstration-scale simulations
- Develop & incorporate energy and thermal efficient exascale algorithms

Year 1.5

- Develop and release first version of CMT-nek for microscale simulations
- Develop and release first version of CMT-nek for mesoscale simulations
- CMT-bone development
- CMT-nek in nek5000 repository

Presentation

- Mrugesh Shringarpure and Jason Hackl

Micro simulation - 4000 MPI ranks

11520 elements, LX1=15, 12 million particles, 7680 MPI ranks

3: CMT-bone Development

Goals

- Co-design an exascale code (CMT-nek) for compressible multiphase turbulence
- Perform micro, meso and demonstration-scale simulations
- Develop & incorporate energy and thermal efficient exascale algorithms

Year 1.5

- Develop and release first version of CMT-nek for microscale simulations
- Develop and release first version of CMT-nek for mesoscale simulations
- CMT-bone development
- CMT-nek in nek5000 repository

Presentation

- Mrugesh Shringarpure and Jason Hackl
Goal
- Derive computationally intensive portions of the CMT-nek code and understand its performance, thermal and energy issues

Year 1.5
- Carried out extensive investigation of performance and energy issues for CMT-bone kernels using CHILL and Genetic algorithm
- Optimization of kernels on CPU-GPU hybrid architecture

Presentation
- Tania Banerjee & Sanjay Ranka

---

5: Uncertainty Budget

Goals
- Develop UB as the backbone of the Center
- Unified application of UB for both physics and exascale emulation

Year 1.5
- Identify main uncertainty sources and quantify their contributions to the model uncertainty of the shock tube simulation
- Explore parameter space of the shock tube simulation for UQ and found anomalies (possible model errors) in simulation results
- Help introduce UQ and propagation in the context of exascale emulation
- Develop a UQ tool: Multi fidelity surrogate

Presentation
- Rafi Haftka, Chanyoung Park, Yiming Zhang
6: Exascale Emulation

Goal
- Develop behavioral emulation (BE) methods and tools to support co-design for algorithmic design-space exploration and optimization of key CMT-bone kernels & applications on future Exascale architectures

Year 1.5
- Demonstrated BE methods for device-level calibration & validation (on existing devices) and prediction (on notional devices) for CMT-bone AppBEOb
- Developed proof-of-concept prototype software PDES simulator for device-level studies & lessons learned; experimentation with single-FPGA hardware-accelerated simulator

Presentation
- Herman Lam, Greg Stitt, Nalini Kumar, Carlo Pascoe, Dylan Rudolph, Kasim Alli

Major Goals For Next Year
- Bigger hero run of demonstration problem with more physics
- Mesoscale simulations of SNL, ASU and Eglin experiments
- Microscale simulations of shock + contact over particles
- Development and validation of binary superposition point-particle model
- UB of demonstration problem
- CMT-nek micro and mesoscale simulations
- Energy and thermal aware hybrid implementation of CMT-nek
- NGEE on multiple FPGAs and performance comparison against simulation
Do you have any questions?
Macroscale and Mesoscale Simulations of Compressible Multiphase Turbulence (CMT)

Subramanian Annamalai
Bertrand Rollin

Why is it interesting?
- Explosive-driven particles
- Shock/particle interaction
- Turbulence/particle interaction
- Wide range of length and time scale

Bring predictive capabilities to particle-laden flow simulations

Sarychev peak (source: wikipedia)
Goals

- Hero runs
  - Continue scaling up simulations

- UQ Team – Macro-Meso Team collaboration
  - Perform mesoscale uncertainty quantification

- Macro-Meso integration (multi scale approach)
  - Progressively improve macroscale modeling via mesoscale physics

Outline

Macroscale:
- Hero run

Mesoscale:
- Real gas equation of state (JWL)
- Exploration of a surrogate model for EoS
- Updated numerical scheme: AUSM+up
- Preliminary mesoscale study:
  - 3D SNL’s experiment
  - Wall gap effect SNL’s experiment
  - ASU’s experiment
### Macro/Mesoscale Gantt Chart

<table>
<thead>
<tr>
<th>Task</th>
<th>Year1</th>
<th>Year2</th>
<th>Year3</th>
<th>Year4</th>
<th>Year5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prep for DOE platforms</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Collision/Compaction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Point-Particle model</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adoptive Particles</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T1: Detonation Sensitivity</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T2: ASU Sim</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T3: No-Particle Exp. Sim</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T4: SNL Particle Curtain</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T5: Meso Eglin</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Demonstration Problem</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Eglin (Macro)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Eglin (Meso)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASU</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Outline

**Macroscale:**
- Hero run

**Mesoscale:**
- Real gas equation of state (JWL)
- Exploration of a surrogate model for EoS
- Updated numerical scheme: AUSM+up
- Preliminary mesoscale study:
  - 3D SNL’s experiment
  - Wall gap effect SNL’s experiment
  - ASU’s experiment
CCMT’s Demonstration Problem

X-ray Source

Suspended gauges

Laser illumination

PIV Camera

High-speed Video Camera

Explorer charge + Annulus of particles

X-ray Film Pack
Physical Models – Sources of Error

Detonation phase
- T1: Detonation model
- T2: Multiphase turbulence model
- T3: Thermodynamic & transport model

Compaction/collision phase
- T4: Collision model
- T5: Compaction model

Dispersion phase
- Metal particles
- Explosive material
- Hot, dense, high pr gas
- Shock wave

Experimental apparatus: Frost et al.’s version
- Iron powder (200 μm)
- Spherical glass particles (dry or wet) (120 ± 30 μm)
- Shaving foam to block flash
- Camera view
- Detonator
- Central detonating cord (50 g/m PETN)

Ambient Air

Glass beads 120μm (40% volume fraction)

Solid Particle cloud

Cylindrical Explosive Charge (PETN)
High Speed Video of an Explosive Dispersal of Particles

Courtesy: D.L. Frost

Prediction Metrics

PM-1: Blast Wave Location
PM-2: Particle Front Location
PM-3: Number of Instability Waves
PM-4: Amplitude of Instability Waves
Major Challenges

- “Rigidity” of input for particles ✔
- Rocflu IO incompatible to the size of our cases ❗
- Rocflu memory leak preventing successful run on Vulcan ✔
- Rocflu unadapted post-processing strategy ❗
- Initialization slowness due to bug and extreme memory requirement ✔
- Inability to have a random distribution of particle ❗
- Parallel partitioning ❗
- Limited performance on BG/Q platform ❗

Simulation Description

- Boundary conditions:
  - Outflow at the outer radius
  - Slip walls at the back and front when running a 3D case

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho^H_{He}$</td>
<td>1770 kgm$^{-3}$</td>
</tr>
<tr>
<td>$\rho^Air_{Air}$</td>
<td>1.203 kgm$^{-3}$</td>
</tr>
<tr>
<td>$\rho^P$</td>
<td>2500 kgm$^{-3}$</td>
</tr>
<tr>
<td>$\phi^P_P$</td>
<td>5 %</td>
</tr>
<tr>
<td>$r_0$</td>
<td>3.8 mm</td>
</tr>
<tr>
<td>$r_1$</td>
<td>5 cm</td>
</tr>
<tr>
<td>$L_z$</td>
<td>2 cm</td>
</tr>
<tr>
<td>$N^{cp}$</td>
<td>$5 \times 10^6$</td>
</tr>
</tbody>
</table>
Demonstration Problem: Simulation

- Features:
  - 30 Million computational cells
  - 5 Million computational particles
  - \( r_{\text{max}} = 4.00 \text{m} \)
  - \( t_{\text{max}} = 0.80 \text{ms} \)
  - 4096 cores
Demonstration Problem: Predictions

PM-1 Comparison

- Data from Frost experiment video starts at 0.400 milliseconds
- Possible sources of discrepancy: EoS, initial particle volume fraction, ...

The blast wave is slower in the experiment than in our current simulations

Particle Volume Fraction Effect on Blast Wave

- A Larger Volume Fraction of Particles Slows Down the Blast Wave
Demonstration Problem: Predictions

PM-2 Comparison

- Data from Frost experiment video starts at 1.500 milliseconds
- Possible sources of discrepancy: EoS, initial particle volume fraction, ...

- The particle front is expanding faster in the experiment than in our current simulation

Outline

Macroscale:
- Hero run

Mesoscale:
- Real gas equation of state (JWL)
- Exploration of a surrogate model for EoS
- Updated numerical scheme: AUSM+up
- Preliminary mesoscale study:
  - 3D SNL’s experiment
  - Wall gap effect SNL’s experiment
  - ASU’s experiment
Experimental results show that the detonation temperature of single crystal PETN is approximately 4100 K

A real gas equation of state is needed to improve the simulation
The Jones-Wilkins-Lee (JWL) equations of state are used to predict the pressures of high energy substances (such as explosives) and are:

\[
P_{\text{JWL}}(\rho, e) = A(1 - \frac{\omega}{R_1 \rho})e^{-R_1 \rho} + B(1 - \frac{\omega}{R_2 \rho})e^{-R_2 \rho} + \omega \rho e
\]

\[
T_{\text{JWL}}(P, \rho) = \left(\frac{1}{\gamma \rho \mathcal{C}_\rho}\right)(P - \rho e^{-R_1 \rho} - \rho e^{-R_2 \rho})
\]

where \( V = \frac{\rho_n}{\rho}, \rho_o, A, B, C, R_1, R_2 \) and \( \omega \) are input parameters.

Implementation for mixed cells:
- Mixed air/product cells need to have pressure and thermal equilibrium
- This is done by an iterative process between the equations of state
- Inputs: Mixture density, mixture energy, product mass fraction
- Outputs: Pressure, Temperature

This implementation is costly in terms of computation time on large grids.

Note that the detonation temperature here is much closer to the experimental results.
A surrogate model was developed for use in the pressure/temperature update. The model was developed by generating a sample solution space from 40 input triplets and then fitting a surrogate to pressure and temperature. The initial set of input triplets was found to contain physically unrealizable points. Solution was to take the range of pressures and temperatures we believed were possible during the detonation and work in reverse to get the inputs. Then, a convex hull of the possible input space was created and placed into the model. This model removes the need for iterating between the equations of state. This approach will save more computational time as the number of species in the problem increases. More details on this can be found in the UQ team presentation.

**T9: Numerical Scheme – AUSM+up**

- Discretization Errors
- Geometric Approximation Error
- Detonation Sensitivity Simulation

- Macroscale
  - U/E Quantification
  - ASU Mesoscale Simulations
  - SNL Mesoscale Simulations
  - Eglin Mesoscale Simulations
  - Eglon No-Particle Simulations
  - Other Detonation Microscale Simulation

- Mesoscale
  - ASU Mesoscale Experiments
  - SNL Mesoscale Experiments
  - Eglin Mesoscale Experiments
  - Eglin No-Particle Experiments
  - Other Detonation Microscale Simulation

- Microscale
  - Takayama Experiments
  - Shock Microscale Simulations
  - Eglin Microscale Simulations
  - Eglin Microscale Experiments

- Characterization & Calibration
  - Characterize Particle Bed
  - Characterize Particle Curtain
  - Characterize Particles After Detonation
  - Calibration of Explosion
AUSM+up Motivation

- Single-phase flux computation scheme introduces oscillations/discontinuities where volume fraction changes
- Multiphase flux scheme should be used for multiphase flow simulations
- Formulated Eulerian-Lagrangian framework of the original Eulerian-Eulerian based multiphase AUSM+up scheme

AUSM+up for Eulerian-Lagrangian

Eulerian-Lagrangian formulation requires:

- Balanced gas phase pressure flux:
  - Eulerian-Eulerian balances mixture pressure flux
  - Eulerian-Lagrangian requires balance of the gas phase
  - Particle-based volume fraction calculation is key
- Consistent force and heat transfer terms:
  - Particle force and heat transfer must be fed back to gas
  - Hyperbolicity term for gas must be fed forward to particles
- Dependent variable decoding step:
  - Eulerian-Eulerian assumes fluid-fluid, simultaneously solve for pressure and volume fraction
  - Eulerian-Lagrangian finds volume fraction directly from particles
- Interface speed of sound
Comparison: AUSM+ vs. AUSM+up

1D simulation of Sandia’s Particle Curtain experiment at t=340μs

AUSM+up with improved volume fraction computation is necessary

T4: Shock – particle curtain simulations

Macroscale: Discretization Errors → Macroscale U/E Quantification → T4

Mesoscale: ASU Mesoscale Simulations → ASU Mesoscale Experiments → T2

Microscale: Takayama Experiments → Shock Microscale Simulations → T6

Characterization & Calibration: Characterize Particle Bed → Characterize Particle Curtain → Characterize Particle Bed → Characterize Particles After Detonation → Calibration of Explosion

Page 32 of 176
Mesoscale Validation: The Particle Curtain Problem

SNL Shock Tube – Justin Wagner

Experimental Data

Shock Tube Simulation

- Validation of the models for gas and particles interaction

Prediction Metric

- **Prediction Metric:** The locations of the particle curtain edges at upstream and downstream
**Particle Curtain Simulation**

- **Features:**
  - 10 Million computational cells
  - 1 Million computational particles

---

**Mesoscale Validation: The Particle Curtain Problem**

_SNL Shock Tube – Justin Wagner_
Half Height Gas Curtain problem

Rightmost position of SF6 significantly different between the top and bottom of the shock tube

How does a particle cloud evolve in a similar setting?

The Particle Curtain Problem Near the Wall

High Speed Camera

Pressure
The Particle Curtain Problem Near the Wall

How opaque is the layer of particles near the top wall?

T2: Expansion fan – particles interaction simulations

Discretization Errors
Macroscopic U/E Quantification

ASU Mesoscale Simulations
SNL Mesoscale Simulations
Eglin Mesoscale Simulations
Eglin No-Particle Simulations

Takayama Experiments
Shock Microscale Simulations
Eglin Microscale Simulations
Eglin Microscale Experiments
Other Detonation Microscale Simulation

Characterize Particle Bed
Characterize Particle Curtain
Characterize Particle Bed
Characterize Particles After Detonation

Calibration of Explosion

Geometric Approximation Error
Detonation Sensitivity Simulation
T2: Experimental Setup

ASU Vertical Shock Tube – Heather Zunino

- Vacuum Tank
- Diaphragm
- Support Brace
- Camera Shelf
- Particle Bed

Particle Cloud - Expansion Fan Interaction

- Features: 20% particle volume fraction
Future Work

- Code optimization on BG/Q
- Co-processing (Catalyst)
- Improved particle collision model
- Particle based volume fraction
- Compaction modeling
- Multiphase compressible LES

Do you have any questions?
Microscale Simulations

Siddharth Thakur, Chris Neal, Yash Mehta, Prashanth Shridharan

Microscale Team

Siddharth Thakur (ST)  S. (Bala) Balachandar  Thomas Jackson  Ju Zhang

Christopher Neal  Yash Mehta  Prashanth Sridharan
Goal of Microscale Simulations

- Fully Resolved Simulations of
  - Structured arrangements of particles
  - Random arrangements of particles
  - Moving and deforming particles

- Develop state-of-the-art point-particle models to account for the effects of volume fraction, Mach number, Reynolds number on
  - Unsteady forces
  - Heat transfer
  - Velocity fluctuations

Multiscale Coupling

**Macro scale**
- $> O(10^0)$ particles
- Macro LES of turbulence
- Point-particle approximation

**Continuum scale**
- EOS, Thermodynamic and transport properties, shock Hugoniot
- Atomistic Quantum and MD

**Mesoscale**
- $O(10^0) - O(10^9)$ particles
- Well resolved interface turbulence
- Unresolved particulate turbulence (Meso-LES)

**Microscale**
- $O(1) - O(10^4)$ particles
- Fully resolved, DNS
- Particle-flow mass, momentum and energy coupling models
Key Accomplishments

- Studies of shock-particle interaction
  - Arrays
  - FCC
  - Random

- UQ-guided grid resolution studies for single and multiple particle simulations

- Improvements to current point-particle model for shock-particle interaction

Gantt Organization Chart

<table>
<thead>
<tr>
<th>Task</th>
<th>Year1</th>
<th>Year2</th>
<th>Year3</th>
<th>Year4</th>
<th>Year5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structured Stationary (FCC)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random Stationary</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Random Moving</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deforming Particle</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Point-Particle Model Development</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model Integration into Demonstration Problem</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UQ-Hybrid Surrogate Model</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Catalyst integration</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dakota Simulation Bundling</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Part 1: Structured Arrays

- Single particle simulations
- Limiting cases of FCC simulations
  - Horizontal array of particles
  - Transverse array of particles
- FCC array of particles
- UQ-guided grid resolution studies

Single Particle Simulation $M_s = 1.22$

- Model comparison with DNS, without bow-shock effects

- Point particle model accurately predicts force on a single particle for sub-critical Mach number
**Single Particle Simulation** $M_s = 6$

- Model comparison with DNS, with bow-shock effects present

- Model with numerical fit to include the effect of bow shock for supercritical Mach number

---

**Effect of Particle Spacing: Horizontal Array**

- Array of spherical particles aligned with flow direction

- Similar simulation was done with $M_s = 1.22$
Effect of Particle Spacing: Horizontal Array

- Shock-particle dynamics at various inter-particle spacing
  - Peak drag coefficient increase is influenced by the incident shock Mach number and inter-particle spacing
  - Particles’ peak drag coefficient has a maximum
  - Peak drag ratio drops below one as inter-particle spacing goes to zero

Transverse Array Simulation

- Shock Mach numbers \( M_s = 3.0 \) and 6.0
- Particle spacing \( L/D = 1.2, 1.5, 2.0, 3.0, \) and 4.0
Transverse Array Simulation

Total force on the center particle for $M_s = 3.0$

- Effect of Mach number and the particle spacing on the force
- Force comparison against single particle

FCC Array (2 unit cells) Simulation

Total force on the center particle for $M_s = 6.0$
FCC Array Simulations

- Simulation matrix for studying effects of volume fraction and Mach number on the drag

<table>
<thead>
<tr>
<th>$M_s$</th>
<th>$\varphi = 10%$</th>
<th>$\varphi = 20%$</th>
<th>$\varphi = 30%$</th>
<th>$\varphi = 40%$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5</td>
<td>RUN1</td>
<td>RUN5</td>
<td>RUN9</td>
<td>RUN13</td>
</tr>
<tr>
<td>2.0</td>
<td>RUN2</td>
<td>RUN6</td>
<td>RUN10</td>
<td>RUN14</td>
</tr>
<tr>
<td>3.0</td>
<td>RUN3</td>
<td>RUN7</td>
<td>RUN11</td>
<td>RUN15</td>
</tr>
<tr>
<td>6.0</td>
<td>RUN4</td>
<td>RUN8</td>
<td>RUN12</td>
<td>RUN16</td>
</tr>
</tbody>
</table>

FCC Array Simulation

$M_s = 1.5 ; \varphi = 10\%$

$M_s = 6.0 ; \varphi = 40\%$
FCC - Effect of Mach number

- $M_s = 1.5$ and $\phi = 10\%$
- $M_s = 6.0$ and $\phi = 10\%$

- $M_s = 1.5$ shows same trend as horizontal array simulation
- For $M_s = 6.0$ peak force drops after 3rd particle – this is different form horizontal array simulation

FCC - Effect of Volume-fraction

- $M_s = 1.5$ and $\phi = 10\%$
- $M_s = 1.5$ and $\phi = 40\%$

- $\phi = 10\%$ shows same trend as horizontal array simulation
- $\phi = 40\%$ does not show any standard trend for the peak force
FCC – Peak Force

Plane 1
- There is no effect of volume fraction on the peak force for plane 1
- Effect of volume fraction is more prominent for downstream particles

Plane 3
- For supercritical Mach bow shock is formed in front of particles
- Bow shocks causes dissipation of energy for lead shock and subsequent decrease in peak force
FCC Grid Resolution Studies

- Collaborating with UQ team on this work

**Simulation Matrix**

<table>
<thead>
<tr>
<th>Surface Mesh</th>
<th>110,000</th>
<th>82,000</th>
<th>70,000</th>
<th>57,000</th>
<th>43,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume Mesh</td>
<td>RUN1</td>
<td>RUN6</td>
<td>RUN11</td>
<td>RUN16</td>
<td>RUN21</td>
</tr>
<tr>
<td></td>
<td>RUN2</td>
<td>RUN7</td>
<td>RUN12</td>
<td>RUN17</td>
<td>RUN22</td>
</tr>
<tr>
<td></td>
<td>RUN3</td>
<td>RUN8</td>
<td>RUN13</td>
<td>RUN18</td>
<td>RUN23</td>
</tr>
<tr>
<td></td>
<td>RUN4</td>
<td>RUN9</td>
<td>RUN14</td>
<td>RUN19</td>
<td>RUN24</td>
</tr>
<tr>
<td></td>
<td>RUN5</td>
<td>RUN10</td>
<td>RUN15</td>
<td>RUN20</td>
<td>RUN25</td>
</tr>
</tbody>
</table>

- Results will be presented by UQ team

Structured Array - Summary

- Structured arrays give very different results compared to single particles

- There is strong effect of $\phi$ and Mach number

- These simulations provide an insight on effect of Mach number and $\phi$ and help in guiding random particle simulations

- Grid resolution studies for structured arrays will serve as benchmark for future simulations
Part 2: Random Particle Packs

Key Components

- Generate 3D Voronoi tessellations for the particles in the random particle packs to estimate local volume fraction & identify nearest neighbors

- Random particle pack simulations for different volume fraction & Mach number

- Framework for extracting information from the force & flow field data from the random particle pack simulations
Random Pack Tessellation

- **Task**: Voronoi tessellation of random particle packs
- Commonly used for granular flow simulation analysis
  - Local volume fraction estimate
  - Nearest neighbors identification

Random Pack Tessellation

- Global volume fraction – 20%
- 400 spherical particles

Random Pack Simulations

- **Task**: Perform 3D multi-particle simulations with randomly packed spherical particles

Mach 3.0 & 10% Volume Fraction

<table>
<thead>
<tr>
<th>Volume Fraction</th>
<th># Particles</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>200</td>
</tr>
<tr>
<td>20</td>
<td>300</td>
</tr>
<tr>
<td>30</td>
<td>400</td>
</tr>
<tr>
<td>40</td>
<td>500</td>
</tr>
</tbody>
</table>
Random Pack Data Analysis

- **Task:** Develop a framework for extracting data from the flow field results of the random particle simulations
- Challenging to extract relevant information from data-rich 3D simulations
- Paraview was used to extract transverse averaged flow field quantities

**Transverse Averaging Planes**

- Global volume fraction – 10%
- 200 spherical particles

Transverse averages allow us to see the average streamwise variations during shock interaction

The shock front can be tracked as it moves through the particle bed using transverse averages

The pack affects the behavior of the planar shock as it moves through the particle pack
Random Pack Data

- Mach 3.0 shock traveling through the particle beds
  - $\phi = 10\%$
  - $\phi = 15\%$
  - $\phi = 20\%$
  - $\phi = 25\%$

Random Pack Data

- Particle drag coefficients & peak drag coefficients within particle bed
Random Pack Data

- Mach 3.0 shock traveling through particle bed

\[ \phi = 10\% \]
\[ \phi = 15\% \]
\[ \phi = 20\% \]
\[ \phi = 25\% \]

Random Particle Packs - Summary

- Forces on particles are chaotic
- Volume fraction of bed does have a noticeable effect on shock
- Methods for descriptive analysis of data are still being produced
- There is an obvious need for information from simpler problems in order to understand the random particle packs
Part 3: Point Particle Modeling

\[ F(t) = ? \]

\[ F(t) = F_{UD}(t) + F_D(t) \]

Point Particle Model

\[ F_{UD}(t) = F_{SG}(t) \]

\[ F_D(t) = F_{QS}(t) + F_{IU}(t) + F_{VU}(t) \]

- \( F_{SG} \) = Stress Gradient
- \( F_{QS} \) = Standard Drag
- \( F_{IU} \) = Inviscid Unsteady
- \( F_{VU} \) = Viscous Unsteady

Current Model (Generalized Faxen Theorem)

\[ F_{UD}(t) = \nabla \left( \frac{\rho u^V}{\partial t} \right) \]

\[ F_D(t) = 6\pi \mu a \left( \frac{u^S}{u} \right) + \nabla \int_{t_{\epsilon}}^{t} \left( K_{iu} \left[ \frac{\partial u^V}{\partial t} \right] \frac{C_0}{a} \right) d\epsilon \]

\[ K_{iu} = \exp \left[ -\mu \left( t - \epsilon \right) \frac{C_0}{a} \right] \cos \left[ -\left( t - \epsilon \right) \frac{C_0}{a} \right] \]
Point Particle Model

- Force on a spherical particle for shock Mach number $M_s = 1.22$

Point Particle Model

- We have state-of-the-art model to predict force on a single particle
- This model takes averaged velocity and density as an input
- How to obtain the force on a particle, when it is surrounded by other particles?
- We have a novel approach – "Binary Superposition"
Binary Superposition

- As shock travels over $\text{(1)}$, it diffracts the flow field
- $R$ sees the flow diffracted by $\text{(1)}$ + shock
- $R$ diffracts the both these flow fields
- $F_R = F_{\text{shock}} + F_{1-R}$

Simulation for $M_s = 1.22$; $L/d_p = 1.5$
- $F_R = F_{\text{shock}} + F_{1-R} + F_{2-R} + \cdots$
Virtual Particle Simulation

\[ F_{1-R} \]

Virtual Particle Simulation

\[ F_{2-R} \]
Binary Superposition - Drag

- \( F_R = F_{\text{shock}} + 4 \times F_{1-R} + 4 \times F_{2-R} \)

CCMT

Binary Superposition - Drag

- \( F_R = F_{\text{shock}} + 3 \times F_{1-R} + 2 \times F_{2-R} \)

CCMT
Binary Superposition - Drag

\[ F_R = F_{\text{shock}} + 2 \times F_{1-R} + 1 \times F_{2-R} \]

Binary Superposition – Lift(Z)

\[ F_R = 2 \times F_{1-R} + 1 \times F_{2-R} \]
Point Particle Modeling - Summary

- We have state-of-the-art single particle model
- There is a need to extend this model for super-critical Mach numbers and include volume fraction effects
- Superposition is a novel way to account for effect of volume fraction

Do you have any questions?
Experimental Studies of Gas-Particle Mixtures Under Sudden Expansion

Heather Zunino
Ph.D. Student

Ronald J. Adrian, Ph.D.
Regents' Professor and Ira A. Fulton Professor of Mechanical & Aerospace Engineering

Problem Statement and Goals

- Experimental multi-phase studies involving compressible flow are complicated
  - Air and solid particles may move separately
  - Particles generate turbulence
- Need for a simple 1D flow experiment that can be used for early validation of the computational codes developed by the PSAAP center.
  - Simpler physics involved than the PSAAP capstone experiment
  - Reduce the scatter in current data (*Chojnicki, et al.*)
- Perform experiments on existing shock tube setup
  - Determine improvement points and weaknesses
- Design an improved, simple 1D compressible multi-phase flow shock tube experiment
  - Examine expansion fan, flow structures, turbulence, and instabilities
- Provide data for early-stage validation of computational codes developed by the PSAAP Center
Review Proposed Experiment

- Six foot glass tube
  - Square footprint
    - 6” x 6”
- Particle bed
- Diaphragm
  - Mylar
- High-speed Cameras
  - Schlieren
  - Background Oriented Schlieren
- Measurements
  - Contact line velocity
  - Gas velocity
  - Particle volume concentration
  - Particle interface
- Parameters: particle size and pressure ratio

Simple Test Bed for Early-Stage Code Numerics

Shock Tube Experimental Structure Setup
New Shock Tube Alterations

- New location secured to put shock tube
  - Taller
  - May reuse some parts from old shock tube
- Round cross section works well
  - Easier and cheaper to assemble (will allow us to make taller)
- Camera purchase
  - Vision Research v641

<table>
<thead>
<tr>
<th>Vision Research (Phantom) v641</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resolution</td>
</tr>
<tr>
<td>Frame rate (at max resolution)</td>
</tr>
<tr>
<td>HD resolution (1080p)</td>
</tr>
<tr>
<td>HD resolution (720p)</td>
</tr>
<tr>
<td>Max frame rate</td>
</tr>
<tr>
<td>Bits</td>
</tr>
<tr>
<td>Internal memory</td>
</tr>
<tr>
<td>High G (impact survivability)</td>
</tr>
<tr>
<td>FAST data offload to laptop</td>
</tr>
<tr>
<td>Warranty</td>
</tr>
<tr>
<td>Resolution at 10,000 fps</td>
</tr>
<tr>
<td>ISO Sat</td>
</tr>
</tbody>
</table>

Schlieren
<Schlieren Video>

Background Oriented Schlieren
Particle Images from PIV Experiment

FFT of Particle Images [ImageJ]

Autocorrelation map of Particle Images [Matlab]

X = rand(600) [Matlab]

FFT of Random Noise [ImageJ]

Autocorrelation map of Random Noise [Matlab]
Broad peak for random noise autocorrelation

• Xcorr2 doesn’t apply periodic boundary conditions
  • Need to manually code the autocorrelation, to get a narrower peak (which will be more representative of the background)
  • Because the features are very small in the random noise pattern, the autocorrelation yields a very broad peak
    • The peak is more associated with the area used to correlate the images than the actual pattern
    • I need to add periodic extensions, so the peak will be more associated with the patterns (or non-pattern) in the random noise, rather than the fact that the features are small.

• See next slide for example

Autocorrelation should be a small peak

Plain white image  Autocorrelation yields broad peak
Random Particles FFT with periodic extension

Random Noise FFT with periodic extension
<Background Oriented Schlieren Video>
Particles

<Background Oriented Schlieren Video>
Random Noise
### Theoretical Calculations

#### Expansion Fan (air only)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length of p4 region, L. 4</td>
<td>0.1 m</td>
</tr>
<tr>
<td>gamma</td>
<td>1.4</td>
</tr>
<tr>
<td>( p_1 ) (kPa)</td>
<td>1</td>
</tr>
<tr>
<td>( p_2 ) (kPa)</td>
<td>15.80620265</td>
</tr>
<tr>
<td>( p_3 ) (kPa)</td>
<td></td>
</tr>
<tr>
<td>( p_4 ) (kPa)</td>
<td>101</td>
</tr>
<tr>
<td>( p_4/p_1 )</td>
<td></td>
</tr>
<tr>
<td>M. shock, table value</td>
<td>3.700393</td>
</tr>
<tr>
<td>speed of sound, ( a_1 ) (m/s)</td>
<td>343</td>
</tr>
<tr>
<td>speed of sound, ( a_2 ) (m/s)</td>
<td></td>
</tr>
<tr>
<td>speed of sound, ( a_3 ) (m/s)</td>
<td>146.9252388</td>
</tr>
<tr>
<td>speed of sound, ( a_4 ) (m/s)</td>
<td>343</td>
</tr>
<tr>
<td>Contact line velocity, ( u_p ) (m/s)</td>
<td>-980.373806</td>
</tr>
</tbody>
</table>

#### Reflected Shock Wave (air only)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M , r/(M , r^2 - 1) )</td>
<td>0.579020077</td>
</tr>
<tr>
<td>( y ), goal seek to 0</td>
<td></td>
</tr>
<tr>
<td>( U , \text{reflected} )</td>
<td>320.9035738 [m/s]</td>
</tr>
<tr>
<td>( U , \text{shock} )</td>
<td>1269.14778 [m/s]</td>
</tr>
<tr>
<td>Length of p1 region, L1</td>
<td>3 [m]</td>
</tr>
<tr>
<td>Time before reflection</td>
<td>0.011712394 [s]</td>
</tr>
</tbody>
</table>

\( t = 0 \) is diaphragm rupture

### Variability in Experiments

- Particle bed height
  - Measuring at 90% intensity
    - No particles is 0%
    - All particles is 100%
Conclusions

- New camera purchase
- Secured location for new shock tube
  - Much taller, longer period before re-shock
- Schlieren
- Background Oriented Schlieren
  - Backgrounds
  - Round tube
- Variability in Measurements
  - Height of rising particle bed
CMT-Nek
Status and Progress

Jason Hackl and Mrugesh Shringarpure

CMT-nek team

Paul Fischer, UIUC
Li Lu, UIUC

Jason Hackl, UF
David Zwick, UF

Mrugesh Shringarpure, UF
Goran Marjanovic, UF
Goals

- Exascale code for predictive simulations of compressible multiphase flows at
  - Microscale (resolved flow over individual particles)
  - Mesoscale (point particles and DNS)
  - Macroscale (point particles and LES)
- Spectral accuracy, robust multiphysics capability
- Co-design of code and algorithm selection by simulation on uncertain notional architectures

Y1.5 Accomplishments

- Compressible Navier Stokes solver
- Particle tracking, 1-way coupling
- CMT-nek committed to nek5000 repository
- Mesoscale simulation: Rarefaction through particle bed (ASU experiment)
- Microscale simulation: Force response of a particle in an expansion fan
- CMT-bone development with CS and Exascale team
### CMT-nek development roadmap

<table>
<thead>
<tr>
<th>Task</th>
<th>Year1</th>
<th>Year2</th>
<th>Year3</th>
<th>Year4</th>
<th>Year5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Euler Solver</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compressible Navier-Stokes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shock Capturing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multiphase Turbulence</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Immersed Boundary Method</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integration with Dakota</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integration with Catalyst</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Other physics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Release</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMTMacro</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMT-bone</td>
<td>B1</td>
<td>B2</td>
<td>B3</td>
<td>B4</td>
<td></td>
</tr>
</tbody>
</table>

### Outline

1. Brief review of CMT-nek framework
2. Accomplishments
   a. Compressible Navier Stokes solver
   b. Microscale: particle in an expansion fan
   c. Particle tracking, 1-way coupling
      • Mesoscale simulation: rarefaction through particle bed
   d. Refinement, upstream Nek integration
   e. CMT-bone development
3. Current and near-future work
1. Brief review of CMT-nek framework

\[
U \equiv \phi_g \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{bmatrix} \rightarrow \frac{\partial U}{\partial t} + \nabla \cdot H_i (U, \nabla U) = R_i
\]

\[
H_i^g = \phi_g \begin{bmatrix} (\rho u)u + p \\ (\rho u)v \\ (\rho u)w \end{bmatrix},
H_i^c = \phi_g \begin{bmatrix} (\rho u)u \\ (\rho u)v + p \\ (\rho u)w \end{bmatrix},
H_i^s = \phi_g \begin{bmatrix} (\rho u)u \\ (\rho u)v \\ (\rho u)w + p \end{bmatrix}
\]

\[
H^d_{i+1, j} = -\phi_g \tau_{ij} H^d_{ij} = G_{ijkl} \frac{\partial U_k}{\partial x_l}
\]

Viscous flux Jacobian

\[
H^e_{ij} = \frac{\partial U_k}{\partial x_l}
\]

Gas-particle coupling

\[
R = -\begin{bmatrix} C_{qp} \\ p \nabla \phi_g + M_{qp} \\ \mu_{ij} (\partial \phi_g / \partial x_j) + \varepsilon_{qp} \end{bmatrix}
\]

\[
\text{Flux} = \text{Convective} + \text{Diffusive}
\]

\[
H_i = H_i^g (U) + H_i^c (U, \nabla U)
\]

\[
\text{Mass}
H_i^g = \phi_g \begin{bmatrix} \rho u \\ \rho v \\ \rho w \end{bmatrix},
H_i^t = 0
\]


1. Review of CMT-nek: nodal DG

\[
\int_{\Omega_e} v \frac{\partial U}{\partial t} dV = \int_{\partial \Omega_e} v (H - H^*) \cdot n dA - \int_{\Omega_e} v (\nabla \cdot H) dV + \int_{\Omega_e} vR dV
\]

nek5000

CMT-nek

Reference element

\[
\forall u \in \chi_h^N
\]

\[
U_m, v, H_{mj}(U, \nabla U), R_m \in \chi_h^N = \left\{ U_h : U_h \approx \bigoplus_{e=1}^{n_c} U_e^N, U_e^N \in \mathbb{R}^N \right\}
\]

Piecwise polynomials

CCMT

1. Review of CMT-nek: nodal DG

\[ \int_{\Omega_e} v \frac{\partial U}{\partial t} \, dV = \int_{\partial \Omega_e} v (H - H^*) \cdot n \, dA - \int_{\Omega_e} v (\nabla \cdot H) \, dV + \int_{\Omega_e} v R \, dV \]

- Approximate all functions \((f=U,v,x,\text{and flux functions})\) by
  - Lagrange polynomials \(L(t)\) of degree \(N=nx1-1\)
  - evaluated at Gauss-Legendre-Lobatto quadrature nodes

\[ f(x) \rightarrow f(r) \approx \sum_{l,m,n=1}^{nx1} f((r_l, s_m, t_n)^T, c) L_l(r) L_m(s) L_n(t) \]

- Integrals become matrix-vector products
  - between vectors of nodal values
  - and the Mass Matrix \(M\)

- Further approximate all integrals by
  - Gaussian quadrature on the GLL points

\[ \int_{\Omega_e} v f \, dV \approx \sum_{l,m,n=1}^{nx1} \omega_l \omega_m \omega_n [v(r) f(r) J(r)] |_{r=(r_l, s_m, t_n)^T} \]

Diagonal \(M！\)

---

1. Review of CMT-nek: nodal DG

\[ v^T \left[ \frac{\partial}{\partial t} (JU) \right] = v^T \left[ M^{-1} \bigotimes_{f=1}^{6} E_f A_f \left[ h_f - h_f^* \right] - Dh + [JR] \right] \]

Fully explicit TVDRK3 of Gottlieb & Shu (1998)

- The quadrature choice
  - tensor-nests finite-difference divergence matrix \(D\)
    - even for deformed elements with curved faces, edges
    - is not exact
- This "variational crime" must be stabilized
  - Implement variational dealising (fixing a mistake)
  - For now, filter Legendre modes via Vandermonde matrix \(V\)

\[ V \text{diag}(\sigma) V^{-1} u_m \]

---


1. Review of CMT-nek: validation

2D compressible vortex flows validated without viscosity

Spectral convergence


2. a Compressible Navier Stokes solver

\[ \int_{\Omega_{e}} \frac{\partial}{\partial x_j} \mathbf{H}^i(x) v(x) dV = \int_{\partial \Omega_{e}} (G_{ijkl} S_{lk} - \mathbf{H}^{S*}) v(x) n_j dA - \int_{\Omega_{e}} \frac{\partial}{\partial x_j} (G_{ijkl} S_{lk}) v(x) dV, \]

Viscous flux of the mth conserved variable in terms of auxiliary variable \( S_i \) satisfying

\[ S_{ij} = \frac{\partial U_i}{\partial x_j} = 0 \quad \int_{\Omega_{e}} v S_{ij} dV = \int_{\Omega_{e}} v \frac{\partial U_i}{\partial x_j} dV + \int_{\partial \Omega_{e}} v (U_i - S^*) n_j dA \]

Central flux \( S^* = (U^*_i + U_i)/2 \) ("+" from neighbor, "-" from neighbor)

\[ H^{S*} = \frac{1}{2} \left( (G_{ijkl} S_{lk})^+ + (G_{ijkl} S_{lk})^- \right) \]
2.a Compressible Navier Stokes solver

2D compressible vortex flows validated with viscosity

- Homentropic initialization, viscosity turned on, decay
  - but first, a low-Mach number radial-velocity pulse

CCMT

Colonius, Lele, Moin (1991) *JFM* 230 dedicated axisym CFD code

2.b Microscale: particle in expansion fan

<table>
<thead>
<tr>
<th>$L_x L_y L_z$ ($d_p$)</th>
<th>elements</th>
<th>nx1</th>
<th>points</th>
<th>CFL</th>
<th>cores</th>
</tr>
</thead>
<tbody>
<tr>
<td>300x100$^2$</td>
<td>24704</td>
<td>15</td>
<td>83M</td>
<td>0.611</td>
<td>480</td>
</tr>
</tbody>
</table>

$C_D = F_X / (1/2 \rho u_\infty^2 \pi R^2)$

$(2a_\infty / d_p) t$
2.c Particle tracking, 1-way coupling

Particle tracking performance
- 1000 elements, 14th order polynomial representation inside the element
- 3,375,000 Degrees of Freedom
- Particles with small response time being swept by $u = (0.15,0,0)$
- Run on Mustang using 960 MPI ranks

Average execution time per time step per MPI rank. All values are in milliseconds

<table>
<thead>
<tr>
<th>Particles</th>
<th>125,000</th>
<th>1,000,000</th>
<th>3,375,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flow solver</td>
<td>Old</td>
<td>New</td>
<td>Old</td>
</tr>
<tr>
<td></td>
<td>32.37</td>
<td>32.08</td>
<td>32.41</td>
</tr>
<tr>
<td>Particle tracking time</td>
<td>45.54</td>
<td>21.46</td>
<td>207.91</td>
</tr>
<tr>
<td>Get computational coordinates</td>
<td>39.36</td>
<td>18.93</td>
<td>173.17</td>
</tr>
<tr>
<td>Send to destination MPI rank</td>
<td>-</td>
<td>0.30</td>
<td>-</td>
</tr>
<tr>
<td>Interpolation</td>
<td>5.17</td>
<td>1.49</td>
<td>33.49</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>105.57</td>
</tr>
</tbody>
</table>
2.c Particle tracking, 1-way coupling

Mesoscale simulation: rarefaction through particle bed

11520 elements, LX1=15, 12 million particles, 7680 MPI ranks
2.d Refinement, upstream Nek integration

- CMT-nek framework refined
  - Adheres to the core design principles of nek5000
  - Consistent design choices makes it easier for existing nek users to start using CMT-nek
- Upstream Nek integration
  - CMT-nek has been folded into nek5000 repo
  - Available for all users to checkout and run CMT-nek
  - Sample CMT-nek cases also added
- Mrugesh and Jason enrolled as developers to ANL’s nek5000 repo
  - CCMT commits CMT-nek updates directly to nek repo

2.e CMT-bone development

- Work with CS team
  - Performance improvement of derivative operation through loop unrolling methods
  - Performance of CMT-nek derivative kernels on GPUs
- Work with Exascale team
  - Development of CMT-bone a representative mini-app
  - Modeling communication in CMT-nek/nek5000
2.e CMT-bone development

**CMT-nek Structure**

- Data structures (Arrays)
  - $O(N^3)$
  - $O(N^2)$
  - $O(N_p)$

- Face data exchange $O(N^2)$
- Particle exchange $O(N_p/N^2)$

**CMT-nek workflow**

- Volume to surface $O(N^2)$
- Surface to volume $O(N^2)$
- Pointwise computation for volume points $O(N^2)$
- Move particles $O(N_p)$
- Relocate particles $O(N_p)$

- Face data exchange $O(N^2)$
- Computation for face points $O(N^2, N_p)$
- Derivative computation volume points $O(N^2)$
- Interpolate to particles $O(N^2, N_p)$
- Distribute point particle force on volume data $O(N^2, N_p)$
3. Current and near-future work

CMT-Nek Solver

- AB3 time integrator
- BC Riemann invariants
- RK 3 time integrator

Filters
- Far field, fringe BC

Viscous BR1
- Dealiasing
- Stable viscous, BR2 or penalty

AUSM+, Central flux
- AUSM +up

Lagrangian point particles (1 way)
- Multi-phase terms (2 way)

Artificial viscosity, Shock capturing
- Real gas effects
- Multi-phase Turbulence
- Collision Physics
- Immersed interface

Do you have any questions?
Exascale
Behavioral Emulation

Principal Investigators:
Alan George, Herman Lam, Greg Stitt

Supported Students:
Doctoral: Kasim Alli, Nalini Kumar, Carlo Pascoe, Dylan Rudolph
Masters: Chris Hajas, Parth Shah
Undergrad: Ting-Hsin Kuo

NSF Center for High-Performance Reconfigurable Computing (CHREC)
ECE Department, University of Florida

Goal*

Develop *behavioral emulation (BE)* methods & tools to support:

- Co-design for algorithmic DSE (design-space exploration)
- Optimization of key CMT-nek kernels & applications

On future architectures, up to Exascale

* Goal stated at AST review May 2015
Integration – How Different Pieces Fit

**Center for Compressible Multiphase Turbulence**

**CCMT**

**Integration**

- **Rocflul**
- **nek5000**

**CMT-nek**

- **Code Development Team**
- **Exascale BE Team**

**CMT-bone**

- Key comp. kernels
- Key comm. patterns

**Behavioral Emulation Co-Design**

- **Modeling & validation of models**
  - On existing architectures for CMT-bone kernels & comm. patterns (benchmarking & interpolation)
  - UQ* team interaction
- **Prediction & DSE**
  - Extend validated models to explore notional & future architectures
  - Algorithmic DSE & optimization for CMT-nek kernels & apps on future architectures
  - UQ* team interaction

**Behavioral Emulation**

- **Platform**
- **Testbed benchmarking & experimentation**

**Algorithmic DSE**

- for CMT-nek on future archs up to Exascale

**Code optimization**

- for CMT-nek on existing archs

**CS Co-Design**

- **Code optimization for CMT kernels**
  - Energy performance tradeoff on hybrid architectures
  - Hardware-software co-optimization
  - Performance optimization under hardware-enforced power bounds
  - Modeling component energy and power
- **Load-balancing algorithms:**
  - Implement load-balancing algorithms for PIC problems in CMT-nek on hybrid multicore architectures.
  - Interact with Exascale and UQ* teams

**Behavioral Emulation Co-Design**

- **Modeling & validation of models**
  - On existing architectures for CMT-bone kernels & comm. patterns (benchmarking & interpolation)
  - UQ* team interaction
- **Prediction & DSE**
  - Extend validated models to explore notional & future architectures
  - Algorithmic DSE & optimization for CMT-nek kernels & apps on future architectures
  - UQ* team interaction

**Algorithmic DSE of CMT-bone on future-gen architectures**

**Behavioral Emulation**

- **Platform**
- **Testbed benchmarking & experimentation**

**Application Design-Space Exploration**

- **CMT-nek**
- **CMT-bone:**
  - key kernels & comm patterns

**Architecture Design-Space Exploration**

- **Notional & Future-gen Systems Exploration**
- **Existing Systems & Architectures**

**Behavioral Emulation Platform**

- **Application BEOs**
  - AppBEOs
- **Architecture BEOs**
  - ArchBEOs

**CCMT**

* BEO – Behavioral Emulation Object
Exascale Behavioral Emulation

<table>
<thead>
<tr>
<th>Task</th>
<th>Year1</th>
<th>Year2</th>
<th>Year3</th>
<th>Year4</th>
<th>Year5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cycle 1</td>
<td>Development of BE methods</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Platform experimentation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cycle 2</td>
<td>Emphasis on scaling &amp; acceleration</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>V1 SW and HW simulators</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cycle 3</td>
<td>Evolution of methods to support new requirements of CCMT teams</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>V2 SW and HW simulators, tools/services</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cycle 4</td>
<td>Explore BE methods to support broader DOE applications</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>V3 SW and HW simulators</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Cycle 1:
- **BE concepts and methods:** App BEOs, Arch BEOs (device level), interpolation techniques for computation
- **Tools:** Prototype evaluation platforms: SMP software (SW) simulator for device-level studies & lessons learned; experimentation with single-FPGA hardware (HW) simulator

Cycle 2:
- **BE concepts and methods:** Emphasis on scaling, communication (synchronization, congestion), and acceleration; focus on CMT-Bone
- **Tools:** V1 SW simulator (leverage production simulators) & V1 HW simulator (scalable design); enable early use of simulators for design-space exploration for CCMT researchers

Cycle 3:
- **BE concepts and methods:** Evolution of methods and techniques to support new requirements of CCMT teams
- **Tools:** V2 SW and HW simulators; libraries of arch & app BEOs; more mature services and tools: management, monitoring, reporting, visualization

Cycle 4:
- **BE concepts and methods:** Evolution of methods and techniques to support new requirements of CCMT teams; began exploration of using behavioral emulation for other key DOE mini-apps and future architectures
- **Tools:** V3 SW and HW simulators

Experimental Testbed: Beyond Cycle 1

Experimental Testbed will:
- Support algorithmic design-space exploration for CCMT researchers
- Provide behavioral emulation within SST ecosystem

CCMT Exascale Behavioral Emulator: Released Version

- SST-based coarse-grained simulator
- Growing libraries of arch & app BEOs
- Services and tools:
  - Management, monitoring, reporting, visualization, …
  - Benchmarking, data collection/mining, …

Evaluation Platforms
- Used to perform R&D and experiments on various aspects of BE
- Promising and stable methods and tools migrate into next platform release
Outline: remainder of presentations

1. BE simulation methods & platform
   - BE methods at scale
   - Progress on SST-based BE simulator

2. UQ for BE
   - Introduction of UQ method for BE
   - Progress & walk-through example

3. FPGA-based BE acceleration
   - Review of first-year FPGA studies
   - NGEEv1* design enhancements
   - Accelerating DSE** using BE

4. Network performance characterization
   - Use existing tools (SST) to analyze performance of key CMT-nek communication routines
   - Lessons learned to guide development of network models in BE simulation platform

*NGEE: Novo-G Exascale Emulator
**DSE: Design-Space Exploration

BE Simulation Methods & Platform

Dylan Rudolph (PhD Student)
Overview

Driving Goals:
Develop BE methods at scale
Find a long-term simulation framework to pair with BE

Term Motivation:
We need to determine the accuracy of BE at scale.
We need to estimate the performance (simulation speed; by extension: maximum simulated size) of BE methods (with SST).

Outline:
- Simulator advancements (new capabilities, refinements)
- Overview of full-scale experiment (application, hardware)
- Comparison of simulated and measured execution times
- Our efforts to integrate BE into SST (progress, methods)
  - Speed comparison of BE-infused SST and SST alone
    - Conclusions and future work

Simulator Advancements

We have added new simulation capability to allow for a greater variety of simulations and results.

New Capabilities:
1) Probabilistic simulation (results now shown as distributions); details in next section
2) Greater software modeling capability (can model more types of applications)

Refinements:
1) Input / output formalization
2) Run-time analysis capability (largely for debugging)
3) Separated front-end (compilation, structure generation) from back-end (simulation) to reduce duplication across evaluation platforms

Note: results shown are executed on single-threaded evaluation platform, but should be identical to production version.
Full-Scale Experiment (overview)

In order to evaluate the speed and accuracy of BE simulations, we need to pick a full-scale test application and architecture.

Application Requirements:
1) Must be fairly representative of CMT or NEK-like applications
2) Must be easy to instrument and do performance evaluation
3) Should be fairly portable and easy to implement in simulation
4) Must run on any reasonable architecture without modification

Architecture Requirements:
1) Must be accessible (reasonable node and time allocations)
2) Must have a known and available network/node configuration
3) Should be sufficiently big for a meaningful ‘full-scale’ test

Full-Scale Experiment: Application

We are using a mini-app which is coarsely representative of CMT or NEK-like applications.

Specifics:
- Uses MPI-level parallelism
- Process count driven by Cartesian grid parameters
- Blocking communication with Cartesian, but not necessarily physical, neighboring processes

Pseudo-Code:
1) initialize
2) make a Cartesian virtual topology
3) main phase:
   - perform some surrogate computation
   - talk to my Cartesian neighbors
4) finalize
**Full-Scale Experiment: Architecture**

**Cab:** Computing cluster at LLNL
- 1296 nodes, 40 TB memory, 2.6Ghz Cores
- Two-level switch InfiniBand QDR network
- Fat-tree-like layout
- Microsecond latencies

**Node Architecture**

<table>
<thead>
<tr>
<th>Xeon-E5-2670</th>
<th>Xeon-E5-2670</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3 4 5 6 7</td>
<td>0 1 2 3 4 5 6 7</td>
</tr>
</tbody>
</table>

**QPI**

(9) InfiniBand QDR
(1) InfiniBand QDR

**Switch 0**

- Node 0
- Node 1
- ... (Nodes X, Y)

**Switch 1**

- ... (Multiple Nodes)

**Big Switch A**

**Big Switch B**

**Full-Scale Experiment: Setup**

We simulate the test application on three different subsets of Cab.
The sizes of the modeled subsets are driven by 3D Cartesian mesh sizes:
- Tiny: $2^3$ mesh (8 processes)
- Small: $4^3$ mesh (64 processes)
- Medium: $6^3$ mesh (216 processes)

We then run the test application on the real Cab machine, and compare simulated versus real execution time.

**Small ($4^3$), Medium ($6^3$) Tests:**

```
<table>
<thead>
<tr>
<th>Node 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xeon-E5-2670</td>
</tr>
<tr>
<td>0 1 2 3</td>
</tr>
<tr>
<td>4 5 6 7</td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>Node 4 (Node 18)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xeon-E5-2670</td>
</tr>
<tr>
<td>0 1 2 3</td>
</tr>
<tr>
<td>4 5 6 7</td>
</tr>
</tbody>
</table>
```

**Tiny ($2^3$) Test:**

```
<table>
<thead>
<tr>
<th>Xeon-E5-2670</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3</td>
</tr>
<tr>
<td>4 5 6 7</td>
</tr>
</tbody>
</table>
```

**InfiniBand QDR**

**Blob-Switch**
Experiment Results: Accuracy ($4^3$)

Small Example: Comparison of simulated and real execution time (histogram of 1000 runs of each)

Observations:
- Mean error of roughly 1%
- Measured distribution is comparatively wide due to unrelated system load
- Measured distribution has higher mean due to unrelated system load
- Cab network appears to be well-characterized by a single-switch model

Experiment Results: Accuracy ($2^3$)

Tiny Example: Comparison of simulated and real execution time (histogram of 1000 runs of each)

Observations:
- Mean error of roughly 1%
- Measured distribution has higher mean due to unrelated system load
- Assorted software and hardware state parameters affect result distributions
- Distribution is not well simulated, but we are not targeting network-less simulations
Experiment Results: Accuracy (6^3)

Medium Example: Comparison of simulated and real execution time (histogram of 1000 runs of each)

Observations:
- Mean error of roughly 1%
- Measured distribution is comparatively wide due to unrelated system load
- Measured distribution has higher mean due to unrelated system load
- Network (compared to small example) is faster and less consistent

SST Integration

We’ve been integrating BE into SST, this is how:

SST Capabilities:
- Framework (core)
- Network Models
- Component Models

BE Influences:
- Abstract Network Definitions
- Abstract Hardware Definitions
- Software Definitions
- Probabilistic Simulation

Final Capability:
- Framework (Simulation Environment)
- Software Definitions
- Probabilistic Simulation
- Abstract Network Models
- Abstract Component Models
SST Integration: Progress / Performance

We are slightly shy of half-way done mapping BE-style simulation on top of SST, but can still estimate performance.

Napkin-Math Methodology:

1) Benchmark things that we’ve completed in the BE-SST integration
2) Compare how long those things took to how long they take in the single-threaded evaluation platform
3) Extrapolate and estimate for a few test-case simulator inputs
   - Expected accuracy: within an order of magnitude, no better
   - Proper estimates will be expected on actual benchmark numbers on BE-SST

How big of a simulation (part count) can we run in under an hour?

100-ULOC*-App (100-ULOC-App)

1) BE single-threaded platform: $10^6$ processes and parts
2) BE on SST (single-threaded): $10^7$ processes and parts
3) BE on SST (multi-threaded): $10^8$ processes and parts
   (This could certainly be Exascale, depending on modeling granularity)

*ULOC: Unrolled (no loops) Lines of Code

Conclusions and Future Work

Conclusions:

- For small-subset Cab prediction, BE is sufficiently accurate (1% error)
- Initial estimates show that we should be able to perform Exascale simulations with BE in a parallel SST environment

Future Work:

- Find a machine which permits us to do larger experiments (tens, hundreds, thousands of nodes)
- Ensure that BE methods still hold at those scales (accuracy-wise)
- Complete the development and integration of BE into SST
Overview

- **Motivation**
  - Quantification of errors and uncertainties is required to establish reliable predictive capabilities

- **Goal**
  - Go beyond our previous V&V efforts to quantify effect that uncertain inputs & model imperfection have on BE
    - Determine variance and confidence intervals for simulation output
    - Analyze output sensitivity to variance in simulation inputs
    - Quantify uncertainty when using calibrated BE models for prediction
Outline of Presentation

- Sources of Uncertainty in BE
- BE as a Stochastic Processes Chain
- Probabilistic Uncertainty Propagation via Monte Carlo Simulation
  - Model BE simulation as non-Markov stochastic process
- Walk-through Example
  - Results
- Conclusions

Sources of Uncertainty in BE (1/2)

- BE: thread-level discrete-event simulation of coarse-grained compute and communication events
  - Comp: modeling time for a collection of local operations to execute on a specified target architecture
  - Comm: modeling time for message of specific size to travel through a target network from one point to another
- Implicit wait and queue “events” used as slack variables to reconcile clocks after comm events
  - Wait: thread is ready to receive before message arrives
  - Queue: message arrives before thread is ready to receive
Sources of Uncertainty in BE (2/2)

- Uncertainty & error introduced due to BE’s coarse-grained abstraction of system details
  - **Comm uncertainties**: network congestion, nondeterministic arbitration or scheduling, ...
  - **Comp uncertainties**: kernel overhead, locality of operands, contention for shared resources, ...
  - **Measurement uncertainties**: sensor noise/resolution, randomization, ...
  - **Model uncertainties**: model inadequacy, surrogate error, ...
  - **Statistical uncertainties**: limited number of test/measurements, experiment variability,...

- Many sources of both *aleatory* and *epistemic* uncertainty present

![Behavior Emulation V&V Process](image)
BE as a Stochastic Processes Chain

- BE simulations comprise multiple thread level simulations executing in parallel
  \( \{t_1, t_2, ..., t_n\} \)

- Thread simulation is ordered sum \( \text{(i.e., not commutative)} \) of its simulated comp & comm events
  \( t_i = e_{i1} + e_{i2} + ... + e_{im} \)
  - Comp events function of surrogate inputs and previous local events
  - Comm events function of surrogate inputs, previous local events, and previous events of other sending/receiving thread
  - Wait/queue timings added to reconcile mismatched send/recv clocks

- Simulated system time is largest simulated thread time
  \( S_{time} = \max(t_1, t_2, ..., t_n) \)

Simple 3-Thread Example

- Example:
  - Thread 0 initializes an array and sends chunks to threads 1 & 2
  - Threads 0-2 perform work on their respective chunks
  - Threads 1-2 communicate modified chunks back to thread 0
  - Threads 0-2 reduce, finalize, and exit
“Truth” vs. Traditional BE Simulation Method

- Example code executed on 3 nodes of LLNL Cab system (1,000 runs)
  - Considered “truth” for our experiment
  - mean: 30.31, sd: 0.23, 90%CI: [30.14, 30.49]
- Traditional BE simulation of example targeted at CAB architecture
  - mean: 29.90
  - relative error between means: -1.35%

Breakdown of Event Timing: 1,000 Runs

Recorded “walks” for example code execution

Traditional BE simulated “walk” of example code execution

Traditional BE Simulation method only takes one of many possible paths
Monte Carlo (MC) Approach to BE

- Traditional BE event timing deterministic
  - i.e., single interpolated value generated from microbenchmark distributions

- How do we capture effect of non-deterministic event ordering
  - Use these distributions as input to MC simulation
  - Generate random walks by sampling from distributions with replacement
  - Analyze resulting output distribution

“Truth” vs. Monte Carlo BE Simulation

- MC-based approach (10,000 walks)
  - Predicted execution: Compare to actual execution:
  
<table>
<thead>
<tr>
<th></th>
<th>Predicted execution</th>
<th>Compare to actual execution</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>30.31</td>
<td>mean 30.31</td>
</tr>
<tr>
<td>median</td>
<td>30.35</td>
<td>median 30.31</td>
</tr>
<tr>
<td>standard dev</td>
<td>0.2365</td>
<td>standard dev 0.0344</td>
</tr>
<tr>
<td>range</td>
<td>[29.18 to 31.38]</td>
<td>range [30.29 to 31.30]</td>
</tr>
<tr>
<td>90% CI</td>
<td>[30.14 to 30.49]</td>
<td>90% CI [30.29 to 30.34]</td>
</tr>
</tbody>
</table>
Breakdown of Event Timing: 10,000 Runs

Monte Carlo-based BE Simulated walks of example code execution

3-way Comparison

Analysis of MC simulation output may help reduce “area discrepancy” of Traditional BE Simulation method
Conclusions

- Developed framework for inserting UQ directly into BE simulator
- Presented results for single set of parameters (e.g., # of nodes, size of messages, ...)
  - Further experiments required
  - UQ with CMT-bone
- Future work:
  - Explore possible improvements to BE models
  - Exploring FPGA-accelerated BE Monte Carlo simulations

Do you have any questions?
Overview

- **Motivation**
  - Behavioral Emulation (BE): coarse-grained, multi-scale simulation approach to balance accuracy, speed, and scalability
  - Is it enough as we approach Exascale?

- **Goal**
  - Complement and accelerate BE approach via FPGA-based reconfigurable computing

FPGA-based BE Acceleration

Carlo Pascoe (PhD Candidate)
Kasim Alli (PhD Student)
Landscape of FPGA-acceleration Studies

Original Project Target
- 1 large, Exascale sim distributed over many FPGAs

NGEEv1 Progress
- 1 small, microscale sim limited to a single FPGA

NGEEv1 Enhancements
- Ongoing improvements to allow for sims at larger scale

NGEEv1 Parameter Sweeps
- Multi-FPGA DSE\* limited to a single simulation per device

Pipelined Design-Space Exploration

Potential Target
- Several large, Exascale sims distributed over many FPGAs

Outline of Presentation

- Review of first-year FPGA studies
- NGEEv1 design enhancements
  - Simulation scaling
  - Increased functionality
- Accelerating DSE using BE
  - Pipelining simulations for higher simulation throughput
- Conclusions
Review of First-Year FPGA Studies

NGEEv1: Current experimental platform for BE-FPGA studies

- Functioning prototype running on single FPGA of Novo-G
- Current core density of 90 for Stratix IV, 256 for Stratix V
  - Each core contains one each of appBEO, procBEO, and commBEO
- Limited optimization (i.e., max-resource implementation)
- Isolated experiments showed ~100x speedup over Year-1 SMP simulator

NGEEv1 Enhancement: Simulation Scaling (1)

Issue 1: appBEO scripts stored as thread-level instruction traces in on-chip block RAMs

- Thread-level instruction traces limit simulation size
  - Finite BRAM per FPGA

Short-term solution: FPGA checkpointing

- Checkpointing instruction added to NGEEv1 ISA*
- Store/load BEO states from within instruction stream
- Reassemble (not resynthesize!) between FPGA checkpoints
- Initial tests look promising

Long-term solution: Shared appBEO streaming

- Reduce total appBEO resources per simulation
- Runtime instruction stream from host

*ISA: Instruction Set Architecture
**NGEEv1 Enhancement: Simulation Scaling (2)**

**Issue 2:** *Current NGEEv1 ISA limiting*
- Necessitates verbose representations of simple code
- Inconsistent with recent developments in BE methodology/functionalities

**Solution:** *Augment existing ISA with spanning subset of new BE control-flow instructions*
- Allows for more succinct representation of thread traces
- Increases effectiveness/desirability of previously discussed checkpointing method
- Negatively impacts FPGA BEO density
  - Ongoing experiments to explore this area-functionality tradeoff

**NGEEv1 Enhancement: Simulation Scaling (3)**

**Issue 3:** *Explicit emulation of target communication fabric*
- Necessitates resynthesis of new circuit when simulating “new” network configuration
- Undesirable for fast DSE

**Possible short-term solution:** *utilize analytical models for communication*
- Explicit routing of messages replaced with models or interpolation
- Single optimized FPGA network for all simulated networks, but ...
  - Possible reduction in simulation accuracy & deviation from software results
  - Experimentation required to assess extent of reduction & deviation

**Long-term solution:** *virtual routing overlaid on explicit routing fabric*
- Single optimized FPGA network to handle all simulated networks, but ...
  - Much more difficult to implement efficiently than analytical models
- Existing network reconfigured when simulating “new” network configuration
CCMT

NGEEv1 Enhancement: Functionality (1)

**Issue 4: HW utilization < 100%**
- app & proc BEOs idle while waiting for comm instruction processing

Possible short-term solution: introduce concept of “BEO hyperthreading”
- Add ability for procBEOs to read from multiple instruction streams
- Switch thread context when stalled for comm events
- Added functionality at expense of added resources
- Experiments to determine if benefits outweigh costs

CCMT

NGEEv1 Enhancement: Functionality (2)

MC* for probabilistic simulation & UQ+ analysis
- Earlier session discussed benefits of MC-based simulation
- Disadvantage: greatly increased simulation time
- Lessen impact with FPGAs

Potential solution(1): Implemented at simulation-level
- Sequentially run same simulation configuration
- Random sampling of distributions implemented with memory initialization file preprocessing & FPGA checkpointing

Potential solution(2): Implemented at event-level
- Random number generators used to sample distributions on FPGA
- Each event executed multiple times per simulation
- Reduces overhead associated with starting/Stopping multiple sims
- Greatly increase per BEO resource requirements

CCMT
Accelerating DSE using BE

Pipelined Simulations

- Investigate acceleration of DSE to complement NGEEv1 efforts, which focused on single-simulation speed up
  - DSE is critical due to need to optimize CMT for different potential Exascale architectures
- Goal is to pipeline more than one simulation to explore a different configuration every cycle
  - After some initial latency, can explore large number of different configurations – fast DSE
- Ideal for an FPGA
  - Pipeline is bounded by resources rather than I/O
  - Multiple FPGAs can be used to explore a vast input space – almost embarrassingly parallel

Pipelined Simulations: Approach

- For a given application (left) extract DFG (middle)
- Construct a pipeline (right) from DFG and implement on FPGA
From DFG to Pipeline

How can we use this to accelerate simulations for DSE?


DSE Pipeline: Example

• Using a series of simulations, explore how different matrix sizes affect performance.
DSE Pipeline: Example

- We start by loading simulation configuration (mat_size) and start time (e0) on each cycle.

<table>
<thead>
<tr>
<th>Sim#</th>
<th>mat_size</th>
<th>e0</th>
</tr>
</thead>
<tbody>
<tr>
<td>sim1</td>
<td>3x3_s1</td>
<td>0</td>
</tr>
<tr>
<td>sim2</td>
<td>5x5_s2</td>
<td>0</td>
</tr>
<tr>
<td>sim1</td>
<td>3x3_s1</td>
<td>0</td>
</tr>
</tbody>
</table>
### DSE Pipeline: Example

#### Sim# | mat_size | e0
---|---|---
lsim3 | 7x7_s3 | 0
lsim2 | 5x5_s2 | 0
lsim1 | 3x3_s1 | 0

---

### DSE Pipeline: Example

#### Sim# | mat_size | e0
---|---|---
lsim7 | 4x4_s7 | 0
lsim6 | 6x6_s6 | 0
lsim5 | 3x3_s5 | 0
lsim4 | 5x5_s4 | 0
lsim3 | 7x7_s3 | 0
lsim2 | 5x5_s2 | 0
lsim1 | 3x3_s1 | 0

---

CCMT
### DSE Pipeline: Example

<table>
<thead>
<tr>
<th>Sim#</th>
<th>mat_size</th>
<th>e0</th>
</tr>
</thead>
<tbody>
<tr>
<td>sim8</td>
<td>3x3_s8</td>
<td>0</td>
</tr>
<tr>
<td>sim7</td>
<td>4x4_s7</td>
<td>0</td>
</tr>
<tr>
<td>sim6</td>
<td>6x6_s6</td>
<td>0</td>
</tr>
<tr>
<td>sim5</td>
<td>3x3_s5</td>
<td>0</td>
</tr>
<tr>
<td>sim4</td>
<td>5x5_s4</td>
<td>0</td>
</tr>
<tr>
<td>sim3</td>
<td>7x7_s3</td>
<td>0</td>
</tr>
<tr>
<td>sim2</td>
<td>5x5_s2</td>
<td>0</td>
</tr>
</tbody>
</table>

Simulation output: e7_s1

### DSE Pipeline: Example

<table>
<thead>
<tr>
<th>Sim#</th>
<th>mat_size</th>
<th>e0</th>
</tr>
</thead>
<tbody>
<tr>
<td>sim9</td>
<td>6x6_s9</td>
<td>0</td>
</tr>
<tr>
<td>sim8</td>
<td>3x3_s8</td>
<td>0</td>
</tr>
<tr>
<td>sim7</td>
<td>4x4_s7</td>
<td>0</td>
</tr>
<tr>
<td>sim6</td>
<td>6x6_s6</td>
<td>0</td>
</tr>
<tr>
<td>sim5</td>
<td>3x3_s5</td>
<td>0</td>
</tr>
<tr>
<td>sim4</td>
<td>5x5_s4</td>
<td>0</td>
</tr>
<tr>
<td>sim3</td>
<td>7x7_s3</td>
<td>0</td>
</tr>
</tbody>
</table>

Simulation output: e7_s2
### DSE Pipelining Challenges

- **Implicit dependency of communication might prevent pipelining**
  - Use analytical models

- **Pipeline max operation may be inaccurate**
  - Use a more sophisticated operation

- **Cases where code cannot be pipelined**
  - Model at higher level of abstraction

- **Simulations may be too large for a single FPGA**
  - Implement resource sharing

- **Configuration parameters causing changes in FPGA circuit, requires long compile times**
  - Leverage virtualization research

---

**Coole, J., & Stitt, G.** *Fast, Flexible High-Level Synthesis from OpenCL using Reconfiguration Contexts.* IEEE Micro, 42-53
Conclusions

- Analysis of Exascale systems will require FPGA-acceleration of BE
- First-year focus on techniques to speedup individual simulations across multiple FPGAs
- Current focus:
  - Techniques to parallelize simulations across multiple FPGAs (parameter sweeping) & within single FPGA (DSE pipelining)
  - Enables rapid DSE exploration to help answer questions about how to optimize CMT apps on potential Exascale architectures
Overview

- **Motivation**
  - Use existing tools to analyze performance of CMT-nek communication routines
    - In parallel with R&D of BE methods and tools
  - Lessons learned to **guide development of network models** in BE simulation platform

- **Goal**
  - Explore performance of CMT-bone comm. routines using SST while BE tools mature
  - Evaluate impact of model granularity on **speed and accuracy of simulations**

Outline of Presentation

- SST Network Modeling Libraries
- Application Modeling in SST
- CMT-bone Simulations using SST
  - Application & machine models
- Results
  - Sensitivity to model parameters
  - SST simulation scalability
  - Design-space exploration
- Conclusions
SST Network Modeling Libraries

- SST has various built-in network models
  - **Ember**
    - coarse-grained endpoint model for traffic patterns to drive lower-level models (Firefly)
  - **Firefly**
    - create network traffic based on comm. pattern in the algorithm to drive physical-layer model (Merlin)
  - **Merlin**
    - low-level flexible router which can be used to connect in different network topologies

- Ember is very suitable for our purposes
  - Supports MPI and SHMEM functions
  - Supports abstraction of low-level details

Application Modeling in SST (Motifs)

- Motifs are coarse-grained representations of app behavior, similar to AppBEOS, that capture interactions between network endpoints
  - Look very much like an MPI program (serial flow)
  - Network endpoints can be cores, devices, nodes, etc.
  - Compute blocks or local operations are delay blocks used to pace the simulation similar to our ProcBEOS

- Ember contains motifs for several commonly used comm. patterns
  - e.g., halo exchanges, MPI collectives, sweeps, etc.
  - We extended motifs library by adding models for CMT-nek comm routines
CMT-bone Simulations using SST (1 of 5)

- For simulations we need:
  1. Motif/abstract application description for CMT-bone
  2. Modeling parameters to describe system
  3. SST configuration file specifying motif parameters

```c
// User parameters - application
uint32_t iterations; // Total no. of timesteps being simulated
uint32_t eltsize; // Size of element (5-20)
uint32_t variables; // No. of physical quantities

// User parameters - machine
int32_t px; // Machine size (no. of nodes in 3d dimensions)
int32_t py;
int32_t pz;
int32_t threads;

// User parameters - mpi rank
uint32_t mpix; // Local distribution of the elements on a MPI rank
uint32_t mpy;
uint32_t mpz;
uint32_t mpi; // Total no. of elements per process (100-10,000)

// User parameters - processor
uint64_t presflops; // no. of FLOPS/cycle for the processor
uint64_t presfreq; // operating frequency of the processor
double m_mean;
double m_stdev;
```

CMT-bone Simulations using SST (2 of 5)

- For simulations we need:
  1. Motif/abstract application description for CMT-bone

```c
double nCompute = n_random->getNextDouble();
emq_compute( evq, nCompute ); // Delay block for compute

// rx/x transfers
// If even: recv rx, send xx, recv -x, send -x
// If odd: send rx, recv xx, send -x, recv -x
if ( myX % 2 == 0 ) {
    if ( sends_pos ) {
        emq_recv( evq, x_pos, x_xfSize, 0, GroupWorld );
        emq_send( evq, x_pos, x_xfSize, 0, GroupWorld );
    }
    else {
        if ( sends_pos ) {
            emq_send( evq, x_pos, x_xfSize, 0, GroupWorld );
            emq_recv( evq, x_pos, x_xfSize, 0, GroupWorld );
        }
    }
}
```
CMT-bone Simulations using SST (3 of 5)

- For simulations we need:
  1. Motif/abstract application description for CMT-bone
  2. Modeling parameters to describe network
  3. SST configuration file specifying motif parameters

```python
networkParams = {
  "packetSize": "2048B",
  "link_bw": "4GB/s",
  "input_latency": "50ns",
  "output_latency": "50ns",
  "flitSize": "8B",
  "buffer_size": "14KB",
}

nicParams = {
  "module": "merlin.linkcontrol",
  "packetSize": networkParams["packetSize"],
  "link_bw": networkParams["link_bw"],
  "buffer_size": networkParams["buffer_size"],
  "txMatchDelay_ns": 100,
  "txDelay_ns": 50,
  "nic2host_lat": "150ns",
}
```

CMT-bone Simulations using SST (4 of 5)

- For simulations we need:
  1. Motif/abstract application description for CMT-bone
  2. Modeling parameters to describe network
  3. SST configuration file specifying motif parameters

```python
numNodes = 0 # numNodes = 0 implies use all nodes on network
numCores = 1

def getNetwork():
    platform = 'default'
    topo = 'torus'
    shape = '2x2x2'
    return platform, topo, shape
```

Page 117 of 176
For simulations we need:
1. Motif/abstract application description for CMT-bone
2. Modeling parameters to describe network
3. Ember configuration file specifying motif parameters

```
def getWorkflow(defaults):
    workflow = []
    motif = dict.copy(defaults)
    motif['cmd'] = "Init"
    workflow.append(motif)
    motif = dict.copy(defaults)
    motif['cmd'] = "CMT3D iterations=10000 elements=10 variables=5 px=16 py=16 pz=32"
    workflow.append(motif)
    motif = dict.copy(defaults)
    motif['cmd'] = "Fini"
    workflow.append(motif)
```

Sensitivity to Model Parameters

- Estimating effect of granularity on simulation accuracy

- **Application setup:**
  - element size=10,
  - iterations=1000

- **Machine setup:**
  - 8x8x8 3D torus,
  - pkt size=2048 B

- **Observations:** As flit size approaches pkt size, simulation estimations become increasingly more inaccurate (~30%)
Scaling SST Simulations

- Speed of SST simulations as size of application grows

![Graph showing the relationship between simulated timesteps and SST execution time for single-threaded simulations.](image)

- **Application setup**: 1000 elements/processor, element size = 10
- **Machine setup**: 512 nodes (8x8x8 torus), bw = 4GB/s, pkt size = 2048B, flit size = 8B
- **Observations**: SST execution time increases linearly with an increase in problem size

Design-Space Exploration (1 of 3)

- Effect of varying element size on application execution time

![Graph showing the relationship between element size and simulated time for varying element sizes.](image)

- **Application setup**: 1000 elements/process, 1000 timesteps (iterations)
- **System setup**: 4x4x4 torus with 1 process per node, bw = 4GB/s, pkt size = 2048B, flit size = 8B
- **Observations**: As expected, app execution time (estimated) increases exponentially with increase in element size
Design-Space Exploration (2 of 3)

- **Effect of varying elements on application execution time**

  **Parameter** = Elements/processor

  **Application setup:** element size=10, 1000 timesteps (iterations)

  **System setup:** 4x4x4 torus with 1 process per node, bw=4GB/s, pkt size=2048B, flit size=8B

  **Observation:** Execution time increases almost linearly with an increase in processor load. Computation is the major contributor to this increase.

---

Design-Space Exploration (3 of 3)

- **Weak scaling**

  **Parameter** = machine size & problem size

  **Application setup:** element size=10, 100 timesteps (iterations)

  **System setup:** 3d torus with 1 process per node, bw=4GB/s, pkt size=2048B, flit size=8B

  **Observation:** As problem size and system size increase, the amount of computation per processor remains the same. Communication time grows fast in the beginning before stabilizing.
Conclusions

- SST, with its built-in network models, is a useful simulation tool
  - Easy to modify granularity
  - Relatively simple software description
- SST execution time scales linearly with size of application being simulated
- In-depth analysis is needed to answer critical load-balancing questions
  - Increasing number of elements processed by a processor has lesser impact on execution time than increasing element size
- Future Work
  - Improve application analysis by improving network and processor calibration
  - Compare scalability and accuracy of SST and BE network models to improve BE models

Do you have any questions?
Student Internships

Chris Hajas (MS)
Chris Neal (PhD)
Carlo Pascoe (PhD)
Nalini Kumar (PhD)

LLNL Internship

Chris Hajas

Mentors: Dr. Scott Lloyd and Dr. Maya Gokhale
Overview

- **Project:**
  - “Emulating In-Memory Data Reorganization for HPC Applications”
  - Accelerate SpMV using emulation framework
  - Explore methods to quickly determine future apps to explore

- **Highlights:**
  - Tours of NIF and Sequoia
  - Many seminars and interactions with different groups in lab
  - Poster session to showcase work
  - Outside events such as rafting, hiking, and exploring the bay area

SpMV Acceleration

- **Motivation:**
  - Emerging memory architectures (e.g. HMC) have logic layer allowing for near memory logic/computation
  - Many apps of interest to DOE exhibit irregular, non-cache friendly memory accesses

- **Goal:**
  - Use custom framework on Zynq FPGA + ARM SoC to emulate Data Reorganization Engine (DRE) logic and integrate with apps using custom API
  - Show that SpMV, a key kernel in many scientific apps, is amenable to data reorganization and provides speed and power improvements when used with our (DRE)
App Amenability Studies

- **Motivation:**
  - Only apps with specific characteristics will benefit from our DRE and modifying these apps can be time consuming in some cases
  - DOE apps are large and complex-- determining whether they would benefit is non-trivial and time consuming

- **Goal:**
  - Develop methods to provide quick insight into whether we should further explore a particular app to use in our emulation framework

Conclusion

- **Contributions**
  - Accelerated SpMV by 1.4x and explored architectural design tradeoffs
  - Analyzed and instrumented various DOE apps/kernels to determine amenability to DRE
  - Researched architectural performance counters and wrote wrapper to help determine amenability to DRE
  - Contributed to paper presented at MEMSYS in October
  - Presenting poster of work at SC15 in Austin in November
National Lab Internship

Christopher Neal
Ph.D. Student

S. Balachandar, Ph.D.
William F. Powers Professor of Mechanical & Aerospace Engineering

Kambiz Salari, Ph.D.
Lawrence Livermore National Laboratory

Overview

- Mentor: Dr. Kambiz Salari

- Internship Highlights
  - Summer seminars
  - NIF & computing facility tours

- Projects
  - Random particle beds
  - Particle wake study
Random Particle Beds

- Motivation – Study the interaction of shockwaves with particle beds
- Random particle bed research aligned with Kambiz’s interests
- Ran simulations for 10%, 15%, 20%, 25% bed volume fractions
- Developed analysis script for estimating shock front location

Streamwise average mach number

Particle Wake

- Grid generation assistance - David Dawson, Ph.D.
- Discussions about simulation setup - Fady Najjar, Ph.D.
- What error is committed when ignoring viscous effects in compressible multi-particle simulations?

- Work is on-going – Looking into multiple particle effect
Summary

- LLNL’s intern program was excellent
- It was great to see the work that LLNL does to advance the mission of stockpile stewardship
- I look forward to returning to the lab next summer

LLNL Internship

Carlo Pascoe

“Enabling CCMT Exploration of Future HPC Architectures with Gem5”

Mentor: Dr. Maya Gokhale
Carlo Pascoe

Motivation

– Behavioral Emulation (BE) framework has mainly targeted
  • Micro-level simulations
  • Simple CMT kernels
  • Existing architectures

– Specifics of model/framework to purely notional architectures remain undetermined

– Determination required before BE exploration of future architectures can progress

Goal

– Evaluate feasibility of leveraging gem5 simulator purposed at generating reliable training data for BE notional architecture models

Approach

– Learn the tool

– Experiment with CCMT kernels on various architecture configurations

– Evaluate performance with several quantitative (e.g., accuracy, speed) and qualitative (e.g., ease of use) metrics

Contributions

– In depth analysis of gem5 and CCMT-nek source code

– Extensive research into AMD piledriver microarchitecture

– Development of custom gem5 cpu/memory models

– Modification of gem5 source for increased functionality

– Integration with model for Micron’s Hybrid Memory Cube

– Setup/execution/evaluation of simulation experiments
Conclusion

- gem5 may be useful for generating training data for BE notional architecture models in select situations with a few caveats:
  - Simulations are slow limiting the number of training points that can be productively captured
  - For our purposes, stock gem5 X86 models are not very good representations of contemporary architectures – some internal modification would be needed
  - Workloads taking advantage of advanced processor features (e.g., complicated branch predictors & prefetchers) will not be modeled well
  - Memory behavior is reasonably accurate assuming correct input and setup, but current models only use a few basic write back cache coherency models with the limit of a single read/write port to cache per cpu core

- For training data generation, if components most necessary for capturing dominant workload behavior are modeled well then simulation runtime error should be acceptable
Overview

- **Lab:** Computer Science Research Institute (CSRI), SNL
- **Mentors:** Dr. Simon Hammond, Dr. Arun Rodriguez
- **Manager:** Dr. Jim Ang
- **Buddy:** Dr. Gwen Voskuilen,

- **Project:**
  
  *Simulating CMT-bone communication routines using high-level network endpoint models*

- **Internship highlights:**
  - Presented a poster and gave a random access talk at Salishan conference on High-speed computing in Oregon in April.
  - Submitted abstract and presented poster at the ModSim workshop in Seattle in August.
  - Attended the Center for Computing Research (CCR), summer seminar series
  - Project report submitted to the CCR Summer Proceedings 2015

---

Scalable Network Simulation using **SST***

We are looking at SST* for supporting scalable network simulation

1. Develop abstract end-point models ‘motifs’ for the various communication routines used in CMT-Nek
   - Identified routines: Nearest-neighbor communication using pairwise exchange, all-to-all using crystal routing, allreduce, bcast etc.

2. Of course we need to validate the simulation results:
   - Full application is too complex and cumbersome to do targeted study, so we developed a mini-app ‘CMTBone’ for in-house use

3. Understand the sensitivity of simulations to the various model parameters
   - Our hope is to reduce the number of component models, parameters, and events being simulated
   - It has to be good enough to provide a first-order approximation of performance which can enable application developers to do some early design space exploration

* Structural Simulation Toolkit
Summary

- Worked with the SST team to understand and evaluate its potential use as a BE platform
- Collaborated with CMT-nek development team to analyze the network behavior of our app
- Created application models for SST that have been used for research in our center
- Several opportunities to discuss and present my research (done at Sandia and UF) to the community
Uncertainty Budget
Validation and Uncertainty Reduction

Chanyoung Park, Yiming Zhang
and Raphael (Rafi) T. Haftka
Department of Mechanical & Aerospace Engineering, University of Florida

Uncertainty Budget Team

- PI: Raphael T. Haftka and Nam-Ho Kim
- Postdoc: Chanyoung Park
- Graduate students
  - Giselle Fernandez (Mesoscale 1-D simulation)
  - Yiming Zhang (Extrapolation)
  - Sam Nili (Mesoscale 1-D and 2-D simulation)
- Visiting PhD student
  - Na Qiu (TBD)
- Undergraduate student
  - Justin T. Mathew (Mesoscale ASU experiment)
Objectives

- In order to validate the prediction capability of the demonstration problem
  - Define measurable quantities of interest as metrics
  - Validate the prediction capability with these metrics
  - Establish appropriate uncertainty quantification and reduction frameworks

- Interactions with the other teams in different disciplines

Accomplishments

- Mesoscale validation
  - Validated 1D/2D mesoscale simulations
  - Estimated the error and the corresponding uncertainty in the predictions of the simulations
  - Quantified uncertainties and calculated uncertainty budget

- Interactions
  - Modeling the feasible region of JWL-EOS
  - Studying the possibility of using surrogate model in JWL-EOS code for efficient computing
  - Defining matrix multiplication computation time measurement considering its variability
  - UQ study of the ASU expansion fan experiment

- Extrapolation
**UB Team**

<table>
<thead>
<tr>
<th>Task</th>
<th>Year1</th>
<th>Year2</th>
<th>Year3</th>
<th>Year4</th>
<th>Year5</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1: Detonation Sensitivity Simulation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T2a: Expansion-Fan ASU Exp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T2b: Expansion-Fan Simulation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T3a: No-particle Explosive Exp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T3b: No-particle Explosive Sim</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T4a: Particle Curtain Exp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T4b: Particle Sim</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T5a: Mesoscale Eglint Exp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T5b: Mesoscale Explosive Sim</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T6: Microscale Eglint Exp</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T6a: Microscale Eglint Sim</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T7: Monoscale Shock Simulations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T8: Post Detonation Particle Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T9: Discretization Error Quantification</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T10: Macroscale Eglint Experiments</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T10: T11: Macroscale Simulations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Exascale /CS**
- Generating Data for Exascale and UQ
- Behavioral Emulation for beyond device level
- Behavioral Emulation for CCMT
- Multi-Fidelity Surrogates
- Extrapolation
- Extreme Events

**CCMT**

**Interactions**

- S. Balachandar
- Thomas L. Jackson
- Bertrand Rollin
- Angela Diggis
- Frederick Ouellet
- Saptarshi Biswas
- Herman Lam
- Dylan Rudolph
- Carlo Pascoe
- Nalini Kumar
- Sanjay Ranka
- Tania Banerjee
- Siddharth Thakur
- Yash A. Mehta
- Christopher Neal
- Micro Simulation
- Macro/Meso Simulation
- Exascale
- Computer Science
- V&V and UQ
- Meso Experiment (ASU)
- Meso Experiment (SNL)
- Ronald Adrian
- Heather Zunino
- Justin Wagner
- CCMT
Mesoscale Validation, UQ and UB

**Simulation**
- Bertrand Rollin (1D, 2D, 3D)
- Angela Diggs (1D)
- Saptarshi Biswas (2D)
- Chanyoung Park
- Giselle Fernandez
- Sam Nili
- Justin T. Mathew

**Experiment**
- Justin Wagner

Meso Experiment (SNL)

Extrapolation

**Behavioral emulation**
- Dylan Rudolph

**Grid resolution study**
- Yash A. Mehta
- Christopher Neal

**Exascale**

**Micro Simulation**

**V&V and UQ**

Yiming Zhang
Chanyoung Park
Other Interactions

JWL-EOS surrogate
Frederick Ouellet
Bertrand Rollin

Other Interactions
Chanyoung Park
Justin Mathew

BEO validation and UQ
Dylan Rudolph
Carlo Pascoe
Nalini Kumar

Macro/Meso Simulation

Exascale

V&V and UQ

Computer Science

Timing model for hybrid computing
Tania Banerjee

Expansion fan UB
Heather Zunino

Mesoscale validation, UQ and UB
- Compensating errors resulted in deceivingly small prediction error
- Model error masqueraded as numerical noise

Extrapolation
- Extrapolation worked with reasonable accuracy for several applications (e.g. convergence of simulation, matrix multiplication)
- Applying knowledge (such as on monotonicity) improved extrapolation accuracy

Other interactions
- The feasible region of JWL-EOS solver needs to be identified and modeled
- Surrogates can help to improve computation speed of JWL-EOS solver
Mesoscale Validation, UQ and UB

- **Goals**
  - Validation of the collision and particle force models
  - Estimating errors in the physics models and the corresponding uncertainties using 1D/2D/3D simulations
  - Quantifying cost for reducing uncertainty (UB)

![Diagram of experiments and simulation](image)

Prediction Metrics

- **Prediction Metrics** are upstream and downstream particle front positions **averaged** over initial particle position and diameter variations

Before impact

After impact

![Graph showing curtain thickness over time](image)

Initial curtain thickness of 2 mm at t=0
UB: Uncertainty Reduction Budget

- UR requires to make validation meaningful (i.e. Useful Failure) for further model improvements in the future
- Uncertainty reduction budget (UB) provides a guideline for finding the most efficient uncertainty reduction approach
- Further uncertainty reduction (UR) is not immediate requirement for the mesoscale simulation

Numerical Model Error Reduction

- The behavior of the diameter line is very different from those of the other groups
- Angela Diggs upgraded the code to a newer version using the scheme AUSM+up
- Improved model substantially reduces error and noise
Sim. Improvement Increased Prediction Error

- AUSM+ of Rocflu was improved to AUSM+up
- Numerical model error reduction revealed the physical errors (i.e. collision and particle force models)
- Apparently small prediction error but possibly large compensating errors

Groups of Runs for Detecting Model Error

- Groups of runs on 4 lines with varying one parameter and varying all parameters
- Curtain thickness (t), particle diameter (D), particle volume fraction (PVF) were considered
- The lines converge to one point (PVF=23%, t=2.4mm, D=110μm)
The numerical noise looking model error and noise looking uncertainty were reduced by upgrading AUSM+ to AUSM+up.

Groups of runs detected the numerical uncertainty in the 1D simulation was huge.

---

**Model Error and Uncertainty Reduction**

- The numerical noise looking model error and noise looking uncertainty were reduced by upgrading AUSM+ to AUSM+up.
- **Groups of runs** detected the numerical uncertainty in the 1D simulation was huge.

---

**Validation and UQ Framework**

- **Goal:** estimate the model error and the corresponding uncertainty.

---
Particle Front Position (PM) with Uncertainty

- Distributions modeling uncertainties
- 95% confidence intervals representing uncertainties

Prediction Metric

Uncertainty in $y_{\text{meas}}$  

Uncertainty in $y_{\text{pred}}$

Prediction Metric

Time

Upstream Front Position
Downstream Front Position
Experiments

PMs and Uncertainties of 1D Simulations

- 1D simulations with AUSM+ and AUSM+up
- Error reduction in AUSM+ scheme increased the prediction error (discrepancy between the UFP prediction and measurement)
- The error in AUSM+ compensated the errors in physics models (e.g. collision model, particle force model, etc.) and the improvement revealed the compensated model errors
PMs and Uncertainties of 1D/2D Simulations

- 1D (AUSM+up) and 2D (AUSM+)
- The discrepancy of 2D is apparently smaller than that of 1D but it should be justified by applying AUSM+up on 2D

Prediction Error and Uncertainty

- Prediction error and the corresponding uncertainty tells us the prediction capability of a simulation and the uncertainty in the evaluation
- Uncertainty reduction clarifies the prediction error

Uncertainty in $y_{\text{meas}}$, Uncertainty in $y_{\text{pred}}$
Prediction Errors and Uncertainties of 1D/2D

- Prediction errors of 1D/2D simulations and 95% CI of the uncertainty
- Note that prediction error is the result of model errors and discretization error

UB and Uncertainty Source Contributions

- Decompose the uncertainty in the prediction error into uncertainty sources and will calculate the cost for reducing the uncertainty sources (UB)
- The uncertainty does not cloud the prediction error for 1D simulation
- However, the uncertainty needs to be reduced for small prediction error with improved models in the future
Extrapolation Overview

- Estimating Convergence of Simulation with Multiple Grid Variables
  - Estimating optimum grid resolution and simulation error of shock simulations

- Method of Converging Lines
  - Estimating performance of future simulation platforms for allocating current resource
  - A benchmark function for behavioral emulation: Matrix Multiplication function

Estimating Discretization Error

- Discretization-based simulations
  - Accuracy depends on grid size
  - Extrapolation allows us to estimate simulation error

- Richardson extrapolation
  - Let $\phi$ be a quantity of interest with the order of error $O(h^q)$
  - With two different meshes $h_1$ and $h_2$, extrapolate at infinity ($h = 0$)
Simulations with Multiple Grid Variables

- 3-D shock simulation with multiple grid variables
  - Volume grid ($N_v$ elements) and surface grid ($N_s$ elements) variables
  - Drag coefficient, $C_D$, on spherical particle
  - Multiple grid variables may have coupled effect on simulation accuracy
  - Simulations were generated by Yash Mehta from Microscale team

3-D Shock Simulations with One Particle

- **Goals**
  - Optimum set of grid variables
  - Error estimation for single simulation
  - Convergence of simulation while refining grid

- 4x4 design of experiments
  - Number of surface elements didn’t affect simulation accuracy noticeably
  - Simulation results look noisy but with monotonic trend
Extrapolation using Constrained Surrogates

- Surrogate maybe an alternative to estimate convergence of simulation
  - Richardson extrapolation may generate meaningless results (e.g. noisy data)
  - Polynomials may be accurate for estimating convergence of simulation based on Taylor series expansion and more robust with noisy data
  - Cubic polynomial was tailored to satisfy the properties of unimodality and monotonicity by controlling the derivatives

Convergence while Refining Volume Grid

- Estimating convergence of simulation using constrained polynomials and Richardson
  - Transformation for number of volume elements: \( h = \frac{1}{\sqrt[3]{N_v}} \)
  - Relative error of extrapolation vary from 0.9% ~1.4%
  - Richardson generated suspicious decrease
  - Estimated simulation error was 83% of the range of data
Method of Converging Lines

- Transform multi-dimensional extrapolation into series of 1-D extrapolations
  - Select lines towards extrapolation point
  - One-dimensional modeling
  - Consistence check between lines
  - Combining lines using Bayesian theory

- Potential benefits
  - Number of samples doesn't depend on dimensionality
  - Validation between lines
  - Obtain physical understanding

Extrapolation for Behavioral Emulation

- Previous Progress
  - Examining Matrix Multiplication function (MMF)
    - Time of $A_{M\times N} \times B_{N\times M}$ generated by Dylan Rudolph from Exascale team
  - Data should be modeled based on quality of data (noise and density)
  - Ridge regression was found to be effective to avoid disastrous error

Extrapolation for Behavioral Emulation
Long-range Extrapolation of Noisy Data

- Extrapolation using method of converging lines
  - Transformation may reveal data pattern: Logarithmic transformation for MMF
  - Line 1 is identified to be outlier mathematically which was verified physically
  - Discrepancy \( d_{ij} = \frac{(p_i - p_j)}{\text{max}(\text{std})} \), \( d_{12} = 2.75 \), \( d_{13} = 3.25 \), \( d_{23} = 0.25 \)

<table>
<thead>
<tr>
<th></th>
<th>Line 1</th>
<th>Line 2</th>
<th>Line 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prediction ( p_1 )</td>
<td>-0.62</td>
<td>-0.73</td>
<td>-0.75</td>
</tr>
<tr>
<td>Prediction variance</td>
<td>( 0.04^2 )</td>
<td>( 0.04^2 )</td>
<td>( 0.03^2 )</td>
</tr>
</tbody>
</table>

- Combined extrapolation has 18% difference with extrapolation point

Summary and Ongoing Work

- **Summary**
  - Proposed constrained polynomials for estimating convergence of simulations especially for which are noisy and complicated
  - Extrapolated a two-dimensional benchmark function of behavioral emulation using the method of converging lines

- **Ongoing work**
  - Evaluating performance of proposed methods
  - Examining more simulations to study grid resolution and error estimation (e.g. shock tube simulation)
  - Testing more benchmark functions of behavioral emulation
Outline for Other Interactions

- JWL-EOS surrogate
- Timing model for hybrid computing
- Preparation for expansion fan UB study

JWL-EOS surrogate

- **Goals:**
  - Modeling the feasible region for the JWL-EOS solver
  - Using surrogates for saving computation time

- Determining the feasible region of JWL-EOS within
  - Mass fraction of explosive [0, 1]
  - Mixture density [1.23, 1770] (kg/m³)
  - Mixture energy [125000, 5700000] (J)

- Fitting surrogates to predict
  - Temperature (K)
  - Pressure (kg/m³)
Methodology of Detecting Feasible Region

- JWL-EOS gives non-physical solutions out of its feasible region (e.g. negative pressure)
- A convex hull of the feasible region was obtained using physical meaningful samples

Surrogate vs. Iteration Solver

- Compared the JWL-EOS iteration solver to the JWL-EOS surrogates
- Both are written by MATLAB
- Computing time for 1000 points
- Ratio of 25 (0.383/0.015)
Surrogate Accuracy

- Relative error: \( \frac{\left| y_{\text{iter}} - y_{\text{surr}} \right|}{y_{\text{iter}}} \)
- Pressure (out of 1000 test points)
  - Rel. error is larger than 1% for small mixture energy and density
  - Red points have more than 10% rel. error

- Temperature (out of 1000 test points)
  - Rel. error is larger than 1% for small mixture density
  - Max abs rel. error is 3%

Timing Model for Hybrid Computing

- **Goal**: Finding computing time evaluation measurement
- Matrix multiplication using GPU is much faster than using CPU
- Communications between CPU and GPU cause variability in total computing time

Communications

- Computation time for \( m \) matrix multiplication operations of matrix size \( n \)
  - \( T_{\text{total}} = T_{\text{comm}} + T_{\text{Comp}} \)
  - \( C_{ik} = B_{ik} A_{il} \)
Computing Time with Variability

- 10000 matrix multiplications (10000 elements)

Tolerance limits for each matrix size: At least 90% of the population of computation time is expected to equal or less than this computation time with 95%

Do you have any questions?
Key Uncertainty Sources

<table>
<thead>
<tr>
<th># Parameters</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Volume fraction</td>
<td>Measurement uncertainty</td>
</tr>
<tr>
<td>2 Particle location</td>
<td>Variability in particle location</td>
</tr>
<tr>
<td>3 Diameter of particle</td>
<td>Variability in diameter</td>
</tr>
<tr>
<td>4 Pressure at driver section $P$</td>
<td>Very small measurement noise</td>
</tr>
</tbody>
</table>

- Particle location and diameter are not measurable
- Average location for random particle locations
**Estimating Discretization Error**

- Need to estimate predictions with
- Three key parameters: (1) grid resolution, (2) temporal resolution and (3) the number of particles per cell
- The effects of (2) and (3) are limited
- Refining grid resolution provokes numerical instability

**Estimating Model Uncertainty**

\[ y_{\text{meas}} + e_{\text{exp}} + e_{\text{meas}} = y_{\text{calc}} + e_{\text{model}} + e_{\text{disc}} + e_{\text{prop}} \]

- Goal is to quantify model uncertainty
- Measurement uncertainty is little

\[ e_{\text{model}} = (y_{\text{meas}} + e_{\text{exp}}) - (y_{\text{calc}} + e_{\text{prop}} + e_{\text{disc}}) \]
Conservative Estimate using UQ

- Conservative prediction of computation time considering variability
  - At least 90% of the population of computation time is expected to equal or less than this computation time

\[ k = t^{-1}\left(\Phi^{-1}(0.9) \sqrt{n, n-1}(0.95)/\sqrt{n}\right) \]

Why UB?

- How much uncertainty can be reduced for given budget?
- The effect of an uncertainty reduction strategy on total uncertainty in discrepancy
UQ Study of ASU Experiments

- **Goal:** To study the expansion pan shocktube experiment in order to identify uncertainty sources while quantifying their magnitudes
- Preparing a white paper describing the experiment procedure and uncertainty sources and the corresponding magnitudes
- Justin T. Matthew has been communicating with Heather Zunino (ASU) in order to learn the experimental set-up and procedure of the shocktube experiments
- Communications allow us to understand the experimental process as well as contribute to a new shocktube design for better UQ
Hardware Software Co-design of CMT-nek Codes
Performance, Energy and Thermal Issues

Tania Banerjee, Mohamed Gadou and Sanjay Ranka
Computer and Information Science and Engineering

Long Term Goals

- Parallelization and UQ of Rocflu and CMT-nek beyond a million cores
- Parallel performance and load balancing
- Single processor (hybrid) performance
- Energy management and thermal issues
Integration

CMT-nek

Rocflu

nek5000

CMT-bone
  - Key comp. kernels
  - Key comm. patterns

Algorithmic DSE
for CMT-nek for future archs up to Exascale

Exascale BE Team

CS Co-Design
- Code optimization for CMT kernels
  - Energy performance tradeoff on hybrid architectures
  - Hardware-software co-optimization
  - Performance optimization under hardware-enforced power bounds
  - Modeling component energy and power
- Load balancing algorithms:
  - Implement load balancing algorithms for PIC problems in CMT-nek on hybrid multicore architectures.
  - Interact with Exascale and UQ teams

Behavioral Emulation Co-Design
- Modeling & validation of models
  - On existing architectures for CMT-bone kernels & comm. patterns (benchmarking and interpolation)
  - UQ team interaction
- Prediction & DSE*
  - Extend validated models to explore notional & future architectures
  - Algorithmic DSE & optimization for CMT-nek kernels & apps on future architectures
  - UQ team interaction

Outline of Talk

- CMT-bone development
- Optimizing CMT kernels on hybrid processors
- Hardware software co-optimization of CMT kernels
- Modeling energy requirements
- Future work
CMT-bone: Key Structures

- Data structures (Arrays)
  - $O(N^3)$
  - $O(N^2)$
  - $O(N_p)$

- Face data exchange $O(N^3)$
- Particle exchange $O(N_p/N^3)$

Work in progress. Joint work with the physics team.
Optimizing CMT kernel on Hybrid Processors

- $\frac{\partial u}{\partial r}(i, j, k) = \sum_{l=1}^{N} A_{il} u_{jk}$
- $\frac{\partial u}{\partial s}(i, j, k) = \sum_{l=1}^{N} B_{il} u_{lk}$
- $\frac{\partial u}{\partial t}(i, j, k) = \sum_{l=1}^{N} C_{il} u_{jl}$

- If $N_x = N_y = N_z = N$
  - Then $B = C = A^T$
- Complexity: $O(N^4)$
- $N$ is typically between 5-25
  - A large number of small matrix multiplications

The derivative computing kernel requires 25-50% of the total solver time of CMT-nek.

CPU-GPU interaction model

- Master-slave
- CPU sends function and operator matrices to GPU, GPU computes derivatives and sends back to CPU

Idle cores utilized to compute derivatives.

Host:
- AMD Opteron 6168
  - 12 cores
  - 1.9GHz clock frequency

Tesla K20c:
- 13 Processors
- 192 Cores
- 48k shared memory
- 64k registers
- 1170 GFLOP/s Peak
- 706MHz clock frequency
Optimization strategy

- On GPU:
  - The derivative operator matrices are only brought in once per block from the device memory to the shared memory. This reduces the number of memory transactions required.
  - The derivative operator matrices are stored in registers instead of shared memory. This reduces the number of accesses to the shared memory.

- On CPU:
  - Using code transformation with proper loop unroll factor and permutation using CHiLL

- Load balancing strategies on CPU and GPU
  - Performance optimal and energy optimal strategies

Possible Code Combinations

Algorithm: \( \text{dudr-4loop} \)

\[
dk = 1, N_x \\
j = 1, N_y \\
i = 1, N_y \\
l = 1, N_x \\
\text{dudr}(l, j, k) = \text{dudr}(l, j, k) + a(l, i) \ast u(l, j, k, ie) \\
\text{enddo} \\
\text{enddo} \\
\text{enddo} \\
\text{enddo}
\]

Number of implementations for \( N_x = N_y = N_z = 10 \)

\[
= 4! \ast 4 \ast 4 = 24 \ast 256 = 6144 \text{ variants}
\]

Total number of variants = 98,240 \((N=10)\)

Total number of variants = 217,728 \((N=20)\)

Question: Can we use a less expensive search technique?
Genetic Algorithm

- We use genetic algorithms to search the exploration space efficiently.
- Individuals represent matrix multiplication variants

```
Input: n

Generate initial population

i=1

Generate algorithm for the ith individual

Compile and run matrix multiplication

Set fitness value of the ith individual (PET)

i < n ?

Sort individuals

Create new generation

Stop ?

Yes

Report the best individual

No

i = i+1
```

Autotuning Framework

3D Matrix multiplication kernel

Genetic Algorithms

Loop transformations

Code generator

Transformed matrix multiplication code

Search Engine

Empirical Performance Evaluation

Optimized matrix multiplication library

Best performing version

Integrate with CMT-nek

Optimized version

CMT-nek

Integrate with CMT-nek

Best performing version

Center for Compressible Multiphase Turbulence
Comparison of GA with Exhaustive Approach

- Comparison of performance by platform

<table>
<thead>
<tr>
<th>Platforms</th>
<th>CMT-bone time (seconds)</th>
<th>Exhaustive Autotuning</th>
<th>GA based Autotuning</th>
<th>Time</th>
<th>Variants</th>
<th>%improve(cmt)</th>
<th>Time</th>
<th>Variants</th>
<th>%improve(cmt)</th>
<th>%lowerPerf</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM BG/Q</td>
<td>5.57</td>
<td>2.54</td>
<td>97716</td>
<td>5.57</td>
<td>2.54</td>
<td>97716</td>
<td>2.58</td>
<td>1124</td>
<td>53.7</td>
<td>1.1</td>
</tr>
<tr>
<td>AMD Opteron</td>
<td>1.81</td>
<td>1.02</td>
<td>97716</td>
<td>1.81</td>
<td>1.02</td>
<td>97716</td>
<td>1.06</td>
<td>1283</td>
<td>41.4</td>
<td>3.9</td>
</tr>
<tr>
<td>AMD Fusion</td>
<td>1.36</td>
<td>0.95</td>
<td>32652</td>
<td>1.36</td>
<td>0.95</td>
<td>32652</td>
<td>0.95</td>
<td>485</td>
<td>30</td>
<td>0.0</td>
</tr>
</tbody>
</table>

- Comparison of energy consumption by platform

<table>
<thead>
<tr>
<th>Platforms</th>
<th>CMT-bone energy (Joules)</th>
<th>Exhaustive Autotuning</th>
<th>GA based Autotuning</th>
<th>Energy</th>
<th>Variants</th>
<th>%improve(cmt)</th>
<th>Energy</th>
<th>Variants</th>
<th>%improve(cmt)</th>
<th>%lowerEnergy</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM BG/Q</td>
<td>292.1</td>
<td>131.7</td>
<td>96 top</td>
<td>292.1</td>
<td>131.7</td>
<td>96 top</td>
<td>135</td>
<td>96 top</td>
<td>53.7</td>
<td>2.5</td>
</tr>
<tr>
<td>AMD Fusion</td>
<td>17.6</td>
<td>11.98</td>
<td>32652</td>
<td>17.6</td>
<td>11.98</td>
<td>32652</td>
<td>12.11</td>
<td>485</td>
<td>31</td>
<td>1.1</td>
</tr>
</tbody>
</table>

Modeling CPU/GPU characteristics

- Modeling helps us gain insight into CPU-GPU characteristics
  - Matrix size N
  - Number of elements, Y

- Helps us balance load using only few runs
  - Alternatives: binary search or autotuning takes large number of experimental runs

- Helps us analyze characteristics of CPU-GPU configurations that are not available to us for direct experimentation

These models will be used as input for simulation by the exascale team.
Modeling Runtime, Power and Energy

**GPU**

<table>
<thead>
<tr>
<th>Runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>T_{gpu} = T_{comp} + T_{comm}</td>
</tr>
<tr>
<td>T_{comp} = 7.17 \times 10^{-11} \times N^{3.76} \times Y</td>
</tr>
<tr>
<td>T_{comm} = 6.14 \times 10^{-9} \times N^3 \times Y</td>
</tr>
</tbody>
</table>

**GPU Power**

<table>
<thead>
<tr>
<th>Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>P_{GPU} = 162.24 \times N^{-0.1}</td>
</tr>
<tr>
<td>Nearly a constant, decreasing very slightly with increasing N.</td>
</tr>
<tr>
<td>Number of memory transactions per unit of data use decreases with N.</td>
</tr>
</tbody>
</table>

**GPU Energy**

<table>
<thead>
<tr>
<th>Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>E_{GPU} = 9.28 \times 10^{-7} \times N^3 \times Y</td>
</tr>
</tbody>
</table>

**GPU Runtime on g GPUs**

<table>
<thead>
<tr>
<th>Runtime on g GPUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>T_{ggpu} = 1/g \times T_{gpu}</td>
</tr>
</tbody>
</table>

**CPU**

<table>
<thead>
<tr>
<th>Runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>T_{cpu} = 1.02 \times 10^{-9} \times N^{4.4} \times Y</td>
</tr>
</tbody>
</table>

**CPU Power**

<table>
<thead>
<tr>
<th>Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>P = P_{memory} + P_{core}</td>
</tr>
<tr>
<td>= 5.95 + 12.21 \times N^{0.3}</td>
</tr>
</tbody>
</table>

**CPU Energy**

<table>
<thead>
<tr>
<th>Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>E_{GPU} = 2.51 \times 10^{-8} \times N^{4.5} \times Y</td>
</tr>
</tbody>
</table>

**CPU Runtime on p CPU cores**

<table>
<thead>
<tr>
<th>Runtime on p CPU cores</th>
</tr>
</thead>
<tbody>
<tr>
<td>T_{pgpu} = 1/p \times T_{cpu}</td>
</tr>
</tbody>
</table>

Load Balancing Results

- **Optimal Energy:** Given a deadline, the GPU should process most of the load with the remaining load being processed by the CPU

<table>
<thead>
<tr>
<th>Deadline(s)</th>
<th>Time(s)</th>
<th>Power(W)</th>
<th>Energy(J)</th>
<th>GPU(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.210</td>
<td>0.210</td>
<td>213</td>
<td>38.6</td>
<td>94.46</td>
</tr>
<tr>
<td>0.190</td>
<td>0.189</td>
<td>279</td>
<td>48.37</td>
<td>85.47</td>
</tr>
<tr>
<td>0.126</td>
<td>0.126</td>
<td>741</td>
<td>93.22</td>
<td>44.56</td>
</tr>
</tbody>
</table>

- **Optimal Performance:** CPU and GPU finish processing at about the same time

<table>
<thead>
<tr>
<th>Deadline(s)</th>
<th>Time(s)</th>
<th>Power(W)</th>
<th>Energy(J)</th>
<th>GPU(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.210</td>
<td>0.2072</td>
<td>213</td>
<td>43.5</td>
<td>91.25</td>
</tr>
<tr>
<td>0.190</td>
<td>0.185</td>
<td>279</td>
<td>51.8</td>
<td>82.83</td>
</tr>
<tr>
<td>0.126</td>
<td>0.1189</td>
<td>807</td>
<td>95.17</td>
<td>43.75</td>
</tr>
</tbody>
</table>
Energy Performance Tradeoff

Circles represent performance and power for a performance optimal implementation
Squares represent performance and power for an energy optimal implementation
Pareto optimal front for hybrid architectures

CPU + GPU Performance

Load = 10000 spectral elements
For CPU-only configurations, runtime drops by a factor of the number of cores
For CPU-GPU configurations, runtime does not drop as profoundly
**CPU + GPU Power Performance**

- CPUs are efficient for smaller matrices
- GPUs are efficient for larger matrices
- CPU+GPU efficiency tend to be independent of matrix size as more CPU cores are added

**Workload ratio**

- With a single CPU core, GPU processes about 4-9 times more load
Hybrid processing of CMT-bone

Communication to other processors

Pool of Spectral Elements

Face data

Summary of work on HMP

- Benchmarked power performance characteristics of the derivative computation kernel on hybrid multi-processors
- Developed empirical models of performance, power and energy consumption of this CMT kernel
- Demonstrated load balancing using performance and energy objectives
- Working on applying these techniques to CMT-bone


Optimizing Hardware Configurations

- L1 Cache Reconfiguration
- L2 Cache Reconfiguration
- DVS of Cores
- DVS of Buses

Reconfigurable Cache

- Associativity Tuning
- Capacity Tuning
- Line Size Tuning

Zhang et al., ACM TECS 2005
Hardware Software Co-optimization

- Hardware parameters
  - L1 cache size – 2K, 4K, 8K
  - L1 line size – 64, 128, 256
  - L1 associativity – 2, 4, 8

- Code Information
  - Partial derivative computation along direction r

- Software parameters
  - Loop permutation
  - Loop unroll factors

- Problem Size (N) – 16

- Number of Code Variations – 4500

- Details of the GEM5 environment
  - Instruction set architecture: X86
  - CPU model: Out-of-order CPU
  - Memory model: Classic, DDR3
  - Clock frequency: 1GHz

Variation in Time

Use GA for finding optimal parameters. Work in progress.
Performance (varying hardware parameters)

Best Time (seconds)

Cache size

Line Size - Associativity

Power for most configurations is nearly the same. Thus, energy requirements are similar.

Modeling Energy Requirements

Challenge: Model the power and energy profiles of the different components
**Approach**

- We have developed a systematic approach of representing the key components as piecewise linear functions.
- This can be developed into an overall linear model that represents the energy requirements.
- Such a linear model has a well defined theory for
  - Modeling the component behavior
  - Developing Uncertainty in the model input and output
  - Design of Experiments for minimizing the number of execution runs

**Piecewise linear approximation**

- Let $f_i(x_i)$ represent the energy requirements of component $i$.
- The range of $x_i$ is evenly split into $k$ intervals.
- Interval $h$ denoted by $[I_h, I_{h+1}]$.
- Within interval $i$, $f_i$ is linearly approximated by $f_{ih} = a_{ij}x_i + b_{ij}$.
- If each component energy only depends on one parameter, the total power consumption $P$ can be approximated by

$$P = \sum_{i=1}^{n} \sum_{h=1}^{k} (a_{ih}x_i + b_{ih}) + P_0$$
Preliminary Experimental Results

- Multiple parameter modeling

\[ P = \sum_{i=1}^{4} f_i(V_i, T_i) + \sum_{i=1}^{4} g_i(H_i) \]

- \( V_i, T_i \) is the supply voltage and current temperature of core \( i \)
- \( H_i \) is cache hit runtime of core \( i \)

Comparison of reconstructed function and actual function.

Preliminary Experimental Results

- Synthetic data

Size of Training Dataset

Average % Error

Error

Size of Training Dataset
**Future Work: CMT-bone Particle Movement**

- Move particles $O(N_p)$
- Relocate particles $O(N_p)$
- Interpolate to particles $O(N_p N_p)$
- Distribute point particle force on volume data $O(N_p N_p)$

Preferential particle clustering
Lagrangian remap

**Different Partitioning Approaches (XGC1)**

- Ensures effective load balancing across regions
- Need to use a spatial indexing data structure like KD-tree to partition triangles
- KD-tree is not very well suited for GPU
- The virtual rectangular grid partitions the mesh into regions
- Load imbalance due to difference in triangle density
- The linear search for triangles can be a bottleneck
Experimental Results

- Mesh from ORNL used for XGC1 benchmarks
  - 1.8 Million triangles
  - Randomly distributed 18 Million particles
  - Level 1 partitioning uses 32 X 32 rectangular grid (regions)
  - NVIDIA Tesla T10 GPU with 4GB global memory, 16k shared memory and 240 computing cores

<table>
<thead>
<tr>
<th>GPU blocks</th>
<th>Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1024</td>
<td>12561.06</td>
</tr>
<tr>
<td>2779</td>
<td>7235.16</td>
</tr>
<tr>
<td>22471</td>
<td>989.88</td>
</tr>
<tr>
<td>33464</td>
<td>428.51</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GPU blocks</th>
<th>Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4096</td>
<td>3111.11</td>
</tr>
<tr>
<td>9216</td>
<td>1366.21</td>
</tr>
<tr>
<td>16384</td>
<td>877.23</td>
</tr>
<tr>
<td>25600</td>
<td>609</td>
</tr>
<tr>
<td>36864</td>
<td>500.92</td>
</tr>
<tr>
<td>50176</td>
<td>427</td>
</tr>
</tbody>
</table>

Future Work: Managing Temperature

Temperature varies on multiple cores

Tilera Processor

[Sarood2011]
Thermal models

- Steady-state thermal model
  - \( T(t) = T_A + G^{-1}P \)
  - Efficient but does not capture transient effects (worst case scenario)

- Transient-state thermal model
  - If the average power of core is \( P \) over a time period \( t \), then the temperature at the end of this period \( T(t) \) is given by:
  - \( T(t) = T_A + e^{-G^{-1}Ct}(T_i - T_A) + G^{-1}(I - e^{-G^{-1}Ct})P \)
  - \( G \) is the thermal conductance matrix
  - \( C \) is the thermal capacitance matrix
  - \( T_A \) is the ambient temperature
  - \( T_i \) is the initial temperature

CS Gantt Chart

<table>
<thead>
<tr>
<th>Task</th>
<th>Year1</th>
<th>Year2</th>
<th>Year3</th>
<th>Year4</th>
<th>Year5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance and energy optimization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>framework applied to CMT-nek</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integrating performance and energy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>optimized kernels in CMT-nek</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Infrastructure for thermal measurements</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>applied on CMT-nek</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Load Balancing algorithms for particulate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>applications</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Algorithms for thermal optimization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>applied to CMT-nek</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PET optimization framework applied to</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CMT-nek</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extend PET optimization framework to</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hybrid Multicore</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extend Load Balancing for Particulate</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Framework to Hybrid Multicore</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PET enabled particulate framework for</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hybrid Multicore</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance evaluation of PET framework</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>on advanced NNSA machines</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance Evaluation of Load Balancing</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Framework on advanced NNSA machines</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exascale/</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UQ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Generating data for Exascale and UQ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>experiments</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Conclusions

- Developed blueprint for CMT-bone development
- Developed novel methods for optimizing CMT kernels on hybrid processors
- Developed hardware software co-optimization approaches for CMT kernels
- Developed methods for modeling energy requirements
- Work in progress on thermal optimization and load balancing for particle in a cell approaches

---

**Do you have any questions?**