Addressing Adaptivity and Scale in Parallel Scientific Simulations

Manish Parashar
The Applied Software Systems Laboratory
ECE/CAIP, Rutgers University
http://www.caip.rutgers.edu/TASSL
(Ack: NSF, DoE, NIH, DoD)

Outline

- Computational Science and Engineering – Trends & Challenges
- Dynamically Adaptive Simulations and Structured Adaptive Mesh Refinement (SAMR)
- Managing Scale and Adaptivity for SAMR Applications
  - Runtime Application Characterization
  - Addressing Spatiotemporal Heterogeneity
  - Addressing Computational Heterogeneity
- Addressing System Issues
- Summary
- Other research projects
Addressing Adaptivity and Scale in Parallel Scientific Simulations

Manish Parashar <parashar@caip.rutgers.edu>

Computational Science & Engineering: Trends & Challenges

- System trends

- Crosscutting challenges - Application, Algorithms, Systems
  - Complex processors, memory, system architectures
  - Very large scales
    - 100 thousand+ processors, millions of threads, ...
  - Heterogeneity
    - Capacity, capability, cost
  - Uncertainty
    - Dynamism, failures

Data acquisition, uncertainty estimation, assimilation, data injection, model adaptation, prediction control, etc.

Computational Research@TASSL

- Conceptual and implementation solutions for solving real-world scientific and engineering problems on very large parallel/distributed systems
- Integration of physical and computational systems
- Key research aspects
  - Algorithms
    - Scalable, asynchronous, latency/failure tolerant, heterogeneity-aware
      - (AI-SAMR, ARXMD, Dispatch, AHMP, HPA/LPA)
  - Programming support
    - Semantically specialized abstractions for dynamic adaptation, coupling, interaction, dynamic data injection
      - (GrACE/DAGH, MACE, Seine, DIOS, Accord)
  - Runtime management
    - Application/physics/system aware management and optimization
      - (Pragma/ARMaDA, HRMS)
  - Computational middleware
    - Interactive monitoring and steering, asynchronous interaction and coordination, end-to-end integration, data sharing, data streaming, collaboration
      - (Discover, Comet, Pawn, Squid, Darts)
### Key Application Domains

- **Plasma edge simulations (SciDAC FSP, Lead NYU)**
  - Coupled core/edge plasma code (PIC + MHD)
  - Large scale coupled simulations, wide area coupling, asynchronous IO, data streaming, in-transit data manipulation

- **Structural biology molecular dynamics (BioMaPS@RU)**
  - Dynamics of ligand-protein complexes using replica exchange
  - Asynchronous replica exchange, latency/failure tolerance, heterogeneity

- **Subsurface geoscience management and control (U of TX, INL, OSU, U of IW, U of AZ)**
  - Oil reservoirs, instrumented oil fields, subsurface contaminant management, waste management, …
  - Coupled multi-physics codes, multiblock formulations with AMR, dynamic data injection, parameter estimation, …

### Key Application Domains

- **Combustion (PPPL, SNL)**
  - Operator-split reaction-diffusion
  - Large scale AMR with spatiotemporal and computational heterogeneity

- **Compressible fluid dynamics (PPPL, SNL, ASCI)**
  - 3-D Richtmyer-Meshkov simulations
  - Large scale AMR with deep localized hierarchies, spatiotemporal heterogeneity

- **Other domains**
  - Numerical relativity, oceanography, seismic modeling, medical informatics (DefeatCancer@WorldCommunityGrid), astrophysics, financial modeling (V@R), …
Addressing Adaptivity and Scale in Parallel Scientific Simulations

Manish Parashar <parashar@caip.rutgers.edu>

Outline

- Computational Science and Engineering – Trends & Challenges
- Dynamically Adaptive Simulations and Structured Adaptive Mesh Refinement (SAMR)
- Managing Scale and Adaptivity for SAMR Applications
  - Runtime Application Characterization
  - Addressing Spatiotemporal Heterogeneity
  - Addressing Computational Heterogeneity
- Addressing System Issues
- Summary

Computational Modeling of Physical Phenomena

- Realistic, physically accurate computational modeling have very large computational requirements
  - E.g., simulation of the core-collapse of supernovae in 3D with reasonable resolution ($500^3$) would require ~10-20 teraflops for 1.5 months (i.e. ~100 Million CPUs!) and about 200 terabytes of storage
- Parallel dynamically adaptive simulations offer an approach for applications with localized features
  - Structured adaptive mesh refinement
- Dynamic adaptivity and scale present significant challenges that limit large scale implementations
  - Spatial, temporal, computational heterogeneity
Structured Adaptive Mesh Refinement (SAMR)

Adaptive Mesh Refinement
- Start with a base coarse grid with minimum acceptable resolution
- Tag regions in the domain requiring additional resolution, cluster the tagged cells, and fit finer grids over these clusters
- Proceed recursively so that regions on the finer grid requiring more resolution are similarly tagged and even finer grids are overlaid on these regions
- Resulting grid structure is a dynamic adaptive grid hierarchy

The Berger-Oliger Algorithm
Recursive Procedure Integrate(level)
If (RegridTime) Regrid
  Step \( \Delta t \) on all grids at level “level”
  If (level + 1 exists)
    Integrate (level + 1)
    Update(level, level + 1)
  End if
End Recursion
level = 0
Integrate(level)

A Selection of SAMR Applications

Blast wave in the presence of a uniform magnetic field – 3 levels of refinement. (ZM9 + GrACE + Cactus, P. Li, NCSA, UCSD)

Mixture of \( \text{H}_2 \) and Air in stoichiometric proportions with a non-uniform temperature field (GrACE + CCA, Jaiddeep Ray, SNL, Livermore)

Richtmyer-Meshkov - detonation in a deforming tube - 3 levels. \( Z=0 \) plane visualized on the right (VTF + GrACE, R. Santaney, CIT)

Multi-block grid structure and oil concentrations contours (IPARS, M. Peszynska, UT Austin)
Computation and Communication in SAMR

- Grids refined in space and time
- Finer grids → more work, typically same load at all grid points
- Parallel SAMR
  - Local computation, intra-level communication, inter-level communication, synchronization overheads, dynamic partitioning and regridding

SAMR Challenges – Heterogeneity in Space and Time

Snapshots of RM3D runtime states.

Workload dynamics
Addressing Adaptivity and Scale in Parallel Scientific Simulations

Manish Parashar <parashar@caip.rutgers.edu>

SAMR Challenges – Computational Heterogeneity

- Computational requirements change in mathematical and topological characteristics – pointwise varying workloads
  - Up to 250:1 relative loads for reactive and diffusive processes
    - Dynamic calibration, load balancing

GrACE: Adaptive Computational Engine for SAMR

- Semantically Specialized DSM
  - Application-centric programming abstractions
  - Regular access semantics to dynamic, heterogeneous, and physically distributed data objects
    - Encapsulate distribution, communication, and interaction
  - Coupling/interactions between multiple physics, models, structures, scales
- Distributed Shared Objects
  - virtual Hierarchical Distributed Dynamic Array
    - Hierarchical Index-Space + Extendible Hashing + Heterogeneous objects
    - Multifaceted objects
    - Integration of computation + data + visualization + interaction
- Adaptive Run-time Management
  - Application and system sensitive management
    - Algorithms, partitioners, load-balancing, communications, etc.
    - Policy-based automated adaptations

Richtmyer Meshkov (3D)
Support for Dynamic Couplings/Interactions

- Multi-numerics
- Multi-physics
- Multi-scale

IPARS Multi-block Oil Reservoir Simulation

Runtime Management of SAMR Applications

- Partitioning/Load-balancing strategies
  - Evolution: Patch, domain, hybrid, meta-partitioners, hybrid, meta-partitioners, ...
    - Many tradeoff (load-balance, computation, communication, speed, etc.)
  - Maximize parallelism, minimize inter/intra level communication, maintain inter/intra level locality, support efficient repartitioning, ...
  - Partitioning/load-balancing strategy depends on the structure of the grid hierarchy and the current application/system state

- Granularity
  - Patch size, AMR efficiency, comm./comp. ratio, overhead, node-performance, load-balance, ...

- Dynamic computational requirements

- Availability, capabilities and state of system resources

Addressing Adaptivity and Scale in Parallel Scientific Simulations

Manish Parashar <parashar@caip.rutgers.edu>

Outline

- Computational Science and Engineering – Trends & Challenges
- Dynamically Adaptive Simulations and Structured Adaptive Mesh Refinement (SAMR)
- Managing Scale and Adaptivity for SAMR Applications
  - Runtime Application Characterization
  - Addressing Spatiotemporal Heterogeneity
  - Addressing Computational Heterogeneity
- Addressing System Issues
- Summary

ARMaDA: Adaptive Application-Sensitive Management of SAMR Applications

1. Cluster and characterize
2. Select a scheme for each clique
3. Repartition and reschedule
Adaptive Hierarchical Multi-Partitioner (AHMP)

• **Approach: Divide-and-Conquer**
  - Identify clique regions and characterize states through clustering
  - Select appropriate partitioner for each clique region to match characteristics of partitioners and cliques
  - Repartition and reschedule within local resource group

• **Hierarchical Partitioning (HPA)**
  - Reduce global communication overheads
  - Enable incremental repartitioning
  - Expose more concurrent communication and computation
  - Addresses spatial heterogeneity

Application Characterization – Identify Cliques

• **Segmentation-based Clustering (SBC)**
  - Formulate a well-structured hierarchy of natural regions (cliques)
  - Identify and characterize the spatial heterogeneity

• **Approach: smoothing and segmentation**
  - Calculate the load density and record its histogram based on space-filling curve (SFC)
  - Find a threshold using the histogram of load density
  - Partition and group sub-regions into several cliques based on the threshold
  - Recursively apply SBC for each clique

### Segmentation-based Clustering (SBC) - Example

Clustering cost is less than < 0.1 % of typical regrid times

### SBC - Clustering Effects

The refinement homogeneity is defined by,

\[
H(I) = \frac{|R^F(I)|}{|R^{total}|} = \frac{\text{Volume of the refined subdomain}}{\text{Volume of the clique } i}
\]

\[
H_{av}(I) = \frac{1}{n} \sum_{i=1}^{n} H(I), \text{ if } |R^F(I)| \neq 0
\]
Cluster Characterization – Octant-based Approach

- Grid structure reflects application runtime states
- Partitioning requirements characterized using its geometry
  - Computation/Communication requirements
    - Computationally-intensive or communication-dominated
    \[ \text{CCratio} = \frac{\sum \text{(Volume of bounding boxes)}}{\sum \text{(Surface area of bounding boxes)}} \]
  - Application Dynamics
    - Rate of change of application refinement patterns
      \[ \text{Dynamics} = \frac{\text{Size of Current state boxes}}{\text{Size of Previous state boxes}} \]
  - Nature of Adaptation
    - Scattered or localized refinements, affecting overheads
- Fast and efficient characterization


Cluster Characterization – Octant-based Approach

- Runtime application monitoring and characterization
  - Computation/communication requirements, application dynamics, nature of adaptation, ..
  - Map partitioners to application state
  - Dynamically select, configure and invoke “best” partitioner at runtime

<table>
<thead>
<tr>
<th>Octant</th>
<th>Scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>pBD-ISP, G-MISP+SP</td>
</tr>
<tr>
<td>II</td>
<td>pBD-ISP</td>
</tr>
<tr>
<td>III</td>
<td>G-MISP+SP, SP-ISP</td>
</tr>
<tr>
<td>IV</td>
<td>G-MISP+SP, SP-ISP, ISP</td>
</tr>
<tr>
<td>V</td>
<td>pBD-ISP</td>
</tr>
<tr>
<td>VI</td>
<td>pBD-ISP</td>
</tr>
<tr>
<td>VII</td>
<td>G-MISP+SP</td>
</tr>
<tr>
<td>VIII</td>
<td>G-MISP+SP, ISP</td>
</tr>
</tbody>
</table>
Hierarchical SAMR Partitioning Framework

- Hierarchical stack of partitioners
  - Maintain locality using space-filling curves
  - Higher-level schemes applied based on local requirements
    - Minimize synchronization costs using LPA
    - Efficiently balance load using BPA
    - Reduce partitioning costs using GPA
    - Combinations of the above …..

SFC: Space-filling Curve
CGDS: Composite Grid Distribution Strategy
HPA: Hierarchical Partitioning Algorithm
LPA: Level-based Partitioning Algorithm
GPA: Greedy Partitioning Algorithm
BPA: Bin-packing based Partitioning Algorithm

Layered design of hierarchical partitioning framework within SAMR runtime engine

ARMaDA/AHMP Operation

Repartition reschedule
Partition schedule
GPA
AHMP
GPA+LPA
AHMP
ALP
ALOC
AHMP
RG1
RG2
RG3
RG4
RG: Resource Group

The load imbalance factor (LIF) is defined by,

\[ LIF(RG-k) = \frac{\max(T_{\text{exec}}^i) - \min(T_{\text{exec}}^i)}{\text{avg of } T_{\text{exec}}^i \text{ in this RG}} \]

\( T_{\text{exec}}^i \): estimated execution time for processor i in RG
ARMaDA/AHMP: Experimental Evaluation

Experiment Setup:
IBM SP4 cluster (DataStar at San Diego Supercomputer Center, total 1632 processors)
SP4 (p655) node: 8 processors (1.5 GHz), memory 16 GB, 6.0 GFlops

Overall Performance

Execution Time for RM3D Application
(1000 time steps, size=256x64x64)

Performance gain
AHMP 30% - 42%
LPA 20% - 29%

GPA: Greedy
LPA: Level-based

Spatiotemporal Heterogeneity in RM3D

- Richtmyer-Meshkov 3-D compressible turbulence
  - Instability occurs at material interface accelerated by a shock wave

- RM3D characteristics
  - Highly localized solution features
    - Small patches and deep application grid hierarchies
  - Unfavorable computation to communication ratios
    - Greater synchronization requirements at higher refinement levels
  - Needs locality, good load balance, reduced synchronization

RM3D detonation in a deforming tube modeled using SAMR

Courtesy: Ravi Samtaney
Evaluation of RM3D Scalability

- RM3D scalability
  - 256, 512, 1024 processors on Blue Horizon
  - 128*32*32 base grid, 4-level hierarchy, 1000 iterations
  - High parallel efficiency (70-83%), good overall performance

```
```

Evaluation of RM3D Scalability

- RM3D SAMR benefits
  - 1024*256*256 resolution and 8000 steps at finest level on 512 processors
  - Around 40% improvement for 5-level hierarchy due to scalable SAMR

```
```
Addressing Adaptivity and Scale in Parallel Scientific Simulations

Manish Parashar <parashar@caip.rutgers.edu>

Outline

• Computational Science and Engineering – Trends & Challenges
• Dynamically Adaptive Simulations and Structured Adaptive Mesh Refinement (SAMR)
• Managing Scale and Adaptivity for SAMR Applications
  – Runtime Application Characterization
  – Addressing Spatiotemporal Heterogeneity
  – Addressing Computational Heterogeneity
• Addressing System Issues
• Summary

Simulations with Heterogeneous Workloads

• Partitioning challenges
  – Different timescales for reactive and diffusive processes
  – Operator-split integration methods in PDEs
  – Highly uneven load distribution as function of space
  – Preserving spatial coupling reduces communication costs
• R-D kernel
  – Ignition of CH₄-Air mixture with 3 “hot-spots”
  – High dynamism, space-time heterogeneity, varying workloads

Addressing Adaptivity and Scale in Parallel Scientific Simulations

Manish Parashar <parashar@caip.rutgers.edu>

---

**Reaction-Diffusion System**

- 2-D methane-air model
  - Operator-split method advances chemistry and transport differently
  - Iterative implicit time-integration scheme for stiff reactive terms
  - Heterogeneous computational load requires non-uniform decomposition
- System solution over global timestep $\Delta t_g$
  \[
  \frac{\partial \phi_i}{\partial t} = \nabla \cdot \frac{P_i \nabla \phi_i}{Q_i} + R(\phi_i) \quad \Rightarrow \quad \frac{\partial \phi}{\partial t} = T(\phi) + R(\phi)
  \]
  - $\phi_i, i = 1, \ldots, N_{species} + 1$ at a grid point is the temperature and the mass fraction of $N_{species}$ chemical species at a given point in space. $R(\phi)$ models the production of heat and chemical species by reversible chemical reactions. Spatial derivatives are approximated using central finite differences.
  - Step 1: Diffusion over $\Delta t_g/2$ ($\Phi^n \rightarrow \Phi'$) – advance $\Phi^n$ using $\Phi'_{i} = T(\Phi)$ to $\Phi'$ over $\Delta t_g/2$ with second-order Runge-Kutta-Chebyshev scheme
  - Step 2: Reaction over $\Delta t_g$ ($\Phi' \rightarrow \Phi''$) – solve $\Phi'_{i} = R(\Phi)$ on point-by-point basis (ODE system solved using CVODE package) to get $\Phi''$
  - Step 3: Diffusion over $\Delta t_g/2$ ($\Phi'' \rightarrow \Phi^{n+1}$) – solve $\Phi'_{i} = T(\Phi)$ over $\Delta t_g/2$ to get $\Phi^{n+1}$ as in Step 1

---

**Dispatch**

- Strategy
  - Calibrate heterogeneity at runtime and determine strategy/trade-off
  - Combines ISP with in-situ weighted global load balancing
- Approach
  - Map computational weights onto current grid structure
  - Generate intermediate workloads using interpolation
  - Compute local loads ("sum-of-parts") and partitioning threshold
  - Domain redistribution preserving application locality

![Dispatch Diagram](image_url)
Addressing Adaptness and Scale in Parallel Scientific Simulations

Manish Parashar <parashar@caip.rutgers.edu>

Evaluation of Dispatch

- **Unigrid evaluation**
  - 8-128 processors on DataStar, 256² and 512² resolution, 200 iterations
  - Execution improved by 12-50%, smaller compute time deviation, reduced sync time, low overheads
- **SAMR evaluation**
  - 2-level: 512² base grid, 3-level: 256² base grid
  - 2-15% performance improvement, granularity affects compute-to-sync
- **SAMR benefits**
  - 512² resolution and 400 steps on finest level for 32 processes
  - 7-42% performance improvement, low compute and sync CV

Evaluation of SAMR benefits

<table>
<thead>
<tr>
<th>Number of processors</th>
<th>Granularity</th>
<th>Homogeneous Time (sec)</th>
<th>Dispatch Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>64</td>
<td>3</td>
<td>1,155.35</td>
<td>1,167.35</td>
</tr>
<tr>
<td>32</td>
<td>4</td>
<td>1,427.11</td>
<td>1,430.36</td>
</tr>
<tr>
<td>128</td>
<td>4</td>
<td>691.33</td>
<td>614.15</td>
</tr>
<tr>
<td>16</td>
<td>5</td>
<td>765.63</td>
<td>664.15</td>
</tr>
</tbody>
</table>

Evaluation of SAMR benefits

Load Balancing and Reaction-Diffusion Compositions

- **Load balancing schemes (at two extremes)**
  - **Blocked distribution**
    - Assumes equal load at each grid point
    - Spatially uniform decomposition along each domain axis
  - **Dispatch strategy**
    - Balances pointwise varying workloads across processors
    - Periodic redistribution to address runtime load heterogeneity

- **Methane-Air models (two compositions)**
  - **Using reduced chemical mechanism**
    - R-D kernel with 25 species and 92 reversible reactions
    - D-R-D splitting, second-order central differences
    - Heterogeneity calibration – pointwise loads vary by order 100-125
  - **Using GRI 1.2 mechanism**
    - CFRFS kernel with 32 species and 177 reversible reactions
    - R-D-R splitting, fourth-order central differences
    - Heterogeneity calibration – pointwise loads vary by factor of 2

Evaluation of Impact of Heterogeneity

- Unigrid evaluation: 64 processors on "Jacquard"
  - R-D kernel: 512^2 resolution, 200 timesteps
  - CFRFS kernel: 500^2 resolution, 20 timesteps
- Heterogeneity analysis
  - CFRFS more compute-intensive → intuitively Dispatch should be better
  - Runtime calibration and compute CV prove otherwise

<table>
<thead>
<tr>
<th>Parameter</th>
<th>R-D Kernel</th>
<th>CFRFS Kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Blocked</td>
<td>Dispatch</td>
</tr>
<tr>
<td>( \mu_{\text{R-D}} )</td>
<td>77.11</td>
<td>81.5</td>
</tr>
<tr>
<td>( \sigma_{\text{R-D}} )</td>
<td>102.14</td>
<td>28.48</td>
</tr>
<tr>
<td>( \mu_{\text{CFRFS}} )</td>
<td>1.327</td>
<td>0.349</td>
</tr>
<tr>
<td>( \sigma_{\text{CFRFS}} )</td>
<td>0.03</td>
<td>9.15</td>
</tr>
<tr>
<td>( \mu_{\text{CV}} )</td>
<td>0.14</td>
<td>15.64</td>
</tr>
<tr>
<td>( \sigma_{\text{CV}} )</td>
<td>0.006</td>
<td>1.709</td>
</tr>
<tr>
<td>( \mu_{\text{R-D}} )</td>
<td>204.42</td>
<td>161.89</td>
</tr>
<tr>
<td>( \sigma_{\text{R-D}} )</td>
<td>102.11</td>
<td>27.77</td>
</tr>
<tr>
<td>( \mu_{\text{CFRFS}} )</td>
<td>0.5</td>
<td>0.172</td>
</tr>
</tbody>
</table>

Outline

- Computational Science and Engineering – Trends & Challenges
- Dynamically Adaptive Simulations and Structured Adaptive Mesh Refinement (SAMR)
- Managing Scale and Adaptivity for SAMR Applications
  - Runtime Application Characterization
  - Addressing Spatiotemporal Heterogeneity
  - Addressing Computational Heterogeneity
- Addressing System Issues
- Summary
Reacting System Sensitive Partitioning

- Cost model used to calculate relative capacities of nodes in terms of CPU, memory, and bandwidth availability
- Relative capacity for node $k$: $C_k = w_p P_k + w_m M_k + w_b B_k$
  - where $w_p$, $w_m$, and $w_b$ are the weights associated with relative CPU, Memory, and Bandwidth availability respectively
- Evaluation
  - Linux based 32 node Beowulf cluster and synthetic load generators
  - RM3D kernel, 128*32*32 base grid, 3 refinement levels, 4 steps regrid
  - 18% improvement in execution time over non-system sensitive scheme

Handle Different Resource Situations

- Efficiency
  - Trade in time (performance) for space (resource)
  - ALP: Trade in time (performance) for space (resource)
- Performance
  - Trade in space (resource) for time (performance)
  - ALOC: Trade in space (resource) for time (performance)
- Survivability
  - ALP: Trade in space (resource) for time (performance)
  - ALOC: Trade in space (resource) for time (performance)
Addressing Adaptivity and Scale in Parallel Scientific Simulations

Manish Parashar <parashar@caip.rutgers.edu>

Outline

• Computational Science and Engineering – Trends & Challenges
• Dynamically Adaptive Simulations and Structured Adaptive Mesh Refinement (SAMR)
• Managing Scale and Adaptivity for SAMR Applications
  – Runtime Application Characterization
  – Addressing Spatiotemporal Heterogeneity
  – Addressing Computational Heterogeneity
• Addressing System Issues
• Summary

Summary

• High performance adaptive simulations can enable accurate solutions of physically realistic models of complex phenomena
  – Scale and adaptivity presents significant challenges
  – Spatial, temporal, computational heterogeneity, dynamism

• Conceptual and implementation solutions for enabling large scale adaptive simulations based on SAMR
  – Computational engines
    • HDDA/DAGH/GrACE/MACE
  – Adaptive runtime management/optimization
    • ARMaDA/HRMS/Dispatch

• Overarching goal
  – Realistic, physically accurate computational modeling and prediction of complex phenomena

• More Information, publications, software
  – www.caip.rutgers.edu/~parashar/
  – parashar@caip.rutgers.edu
Pervasive Computational Ecosystems: Integrating Physical and Computational Worlds

- Knowledge-based, data-driven adaptive and interactive scientific investigation
  - A new generation of scientific and engineering simulations of complex physical phenomena that symbiotically and opportunistically combine computations, experiments, observations, and real-time data
    - The Instrumented Oil Field of the Future (UT-CSM, UT-IG, RU, OSU, UMD, ANL) (NSF)
    - Management of the Ruby Gulch Waste Repository (UT-CSM, INL, OU) (DoE)
    - Adaptive Fusion of Stochastic Information for Imaging Fractured Vadose Zones (OSU, U of IW, RU, U of Az) (NSF)
    - Data-Driven Forest Fire Simulation (U of AZ, RU) (DoE)
    - Data-Driven Management of Coastal Systems (RU) (Pending)

- Challenge: Uncertainty (System, Application, Information)

Data-driven Management of Subsurface Geosystems: The Instrumented Oil Field (with UT-CSM, UT-IG, OSU, UMD, ANL)

Detect and track changes in data during production. Invert data for reservoir properties. Detect and track reservoir changes. Assimilate data & reservoir properties into the evolving reservoir model. Use simulation and optimization to guide future production.

Management of the Ruby Gulch Waste Repository (with UT-CSM, INL, OU)

- Ruby Gulch Waste Repository/Gilt Edge Mine, South Dakota
  - ~ 20 million cubic yard of waste rock
  - AMD (acid mine drainage) impacting drinking water supplies

- Monitoring System
  - Multi electrode resistivity system (523)
    - One data point every 2.4 seconds from any 4 electrodes
  - Temperature & Moisture sensors in four wells


Adaptive Fusion of Stochastic Information for Imaging Fractured Vadose Zones (with U of AZ, OSU, U of IW)

- Near-Real Time Monitoring, Characterization and Prediction of Flow Through Fractured Rocks
Data-Driven Forest Fire Simulation (U of AZ)

- Predict the behavior and spread of wildfires (intensity, propagation speed and direction, modes of spread)
  - Based on both dynamic and static environmental and vegetation conditions
  - Factors include fuel characteristics and configurations, chemical reactions, balances between different modes of heat transfer, topography, and fire/atmosphere interactions.


Conformational Variability of Protein Receptors (with BioMaPS, RU)

- Proteins exist in multiple conformations in solution
- Design of inhibitor drugs should take into account most probable conformations
- Replica Exchange is a powerful method to generate a thermal distribution of conformations
Asynchronous RXMD

- Practical Challenges of parallel/distributed RXMD
  - Requires complex exchange "negotiations"
    - Synchronous with centralized coordination
  - Large systems need many replicas/processors – Scalability
    - Convergence rate decreases with number of replicas
  - Long running – many hours to days
  - Nearest-neighbor exchange strategy is inefficient with many replicas
  - System heterogeneity can have severe impact
    - Cluster based simulations

- Asynchronous formulation and computational engine for asynchronous replica exchange (Comet)
  - Scalable, latency and failure tolerant
  - Allow non-nearest neighbor exchange – dynamically negotiate exchanges
  - Manage heterogeneity


The SciDAC CPES Fusion Simulation Project (FSP)

GTC Runs on Teraflop/Petaflop Supercomputers

Large data analysis

End-to-end system with monitoring routines

Data replication

Data replication

Visualization

User monitoring

Post processing
Addressing Adaptivity and Scale in Parallel Scientific Simulations

Manish Parashar <parashar@caip.rutgers.edu>

Code Coupling and SciDAC CPES Project

CPES Fusion Simulation Project – Requirements

- Scalable coupling of multiple physical models and associated parallel codes that execute independently and in a distributed manner
  - Interaction/communication schedules between individual processors need to be computed efficiently, locally, and on-the-fly, without requiring synchronizations or gathering global information, and without incurring significant overheads on the simulation
  - Data transfers are efficient and happen directly between individual processors of each simulation

- Asynchronous IO:
  - Minimize overhead on compute nodes
  - Maximize data throughput from the compute nodes

- Wide area data streaming and in-transit manipulation
  - Enable high-throughput, low latency data transfer
  - Adapt to network conditions to maintain desired QoS
  - Handle network failures while eliminating data loss
The Team

- **Key Applications Collaborators**
  - Rutgers Univ.
    - R. Levy, S. Garofolini
  - UMDNJ
    - D. Foran, M. Reisse
  - CSM/IG, Univ. of Texas at Austin
  - ORNL, NYU
    - S. Klasky, C.S. Chang
  - CRL, Sandia National Lab., Livermore
    - J. Ray, J. Steensland
  - Univ. of Arizona/Univ. of Iowa, OSU
    - J. –C. J. Yeh, J. Daniels, A. Kruger
  - Idaho National Laboratory
    - R. Versteeg
  - PPPL
    - R. Samtaney
  - ASCI/CACR, Caltech
    - J. Cummings

- **TASSL, CAIP/ECE Rutgers University**
  - Viraj Bhat
  - Sumir Chandra
  - Ciprian Docan
  - Andres Q. Hernandez
  - Nayyan Jiang
  - Zhen Li (Jenny)
  - Vincent Matossian
  - Mingliang Wang

- **Key CE/CS Collaborators**
  - Rutgers Univ.
    - D. Silver, D. Raychaudhuri, P. Meer, M. Bushnell, etc.
  - Univ. of Arizona
    - S. Hariri
  - Ohio State Univ.
    - T. Kurc, J. Saltz
  - GA Tech
    - K. Schwam, M. Wolf
  - University of Maryland
    - A. Sussman, C. Hansen

Thank You!