Towards A Grid Enabled System for Multicomponent Materials Design

Keita Teranishi and Padma Raghavan
Computer Science and Engineering
and
Zi-Kui Liu
Materials Science and Engineering

The Pennsylvania State University
4/21/2004
CCGrid 2004
Chicago, IL
Overview

- Introduction
- Multicomponent-multiscale materials science simulations
  - Components are Al-Cu-Mg-Si phases
  - Scales range from atomistic to macro-structural
- Focus on: Grid enabled system design
  - Front-end: Client-server web-portal
  - Back-end: Globus enabled simulation servers
- Conclusions
Introduction

- Advances in computational materials science in the past few decades for Al-Cu-Mg-Si systems
  - Methods for different modeling steps and their parallel (MPI-based) implementations
  - Methods to perform finite element analysis on simulated-microstructures
- Advances in scientific computing
  - Parallel scalable algorithm/implementations
  - Wide area distributed computing using Grids
  - Extensible markup, definition languages
Matcase: Multiscale Materials Design

- Computational materials design is divided into 4 different steps from atomistic level (Å) to macro-structure (mm)
  - First-principle calculations
  - CALPHAD for phase-diagrams (thermodynamic bulk properties)
  - Phase-field simulation to generate microstructures (precipitate distribution, evolution of meta-stable phases)
  - Finite element analysis to obtain stress-strain curves

Goal: Automating the modeling of Al-Cu-Mg-Si across different simulation scales
**NSF-ITR Project: MATCASE**

- **First-principles calculations**
  - Bulk thermodynamic data
  - Experimental data
  - Bulk thermodynamic database
- **CALPHAD**
  - Interfacial energies, lattice parameters and elastic constants
  - Database for lattice parameters, elastic constants and interfacial energies
- **Phase-field simulation**
  - Plasticity of phases
  - Microstructure in 2D and 3D
- **OOF: Object-oriented finite element analysis**
  - Elasticity of phases
  - Mechanical responses of simulated microstructures
User view

- Users (clients) connect to initiate materials design via web-portal
- Web-portal creates a service to the user and executes remote tasks
- Remote tasks are managed by Globus-enabled services
  - Specifies exact set of simulations needed to compute missing data for a given design
- Our model reuses information in materials databases as much as possible
Connecting Four Steps

Each computational step has different requirements

- Commercial software
- Different programming models
- Different data formats
- Codes for CALPHAD and Phase-field simulation are under development
- Installed in different computers
- Users are not computer scientists!
Connecting Four Steps (continued)

- **Observation:** The output of each step is relatively compact, but some steps are computationally expensive.

- Leave original software (such as VASP) as it is!
  - Natural to use databases in order to provide the linkage between successive computational steps.
  - Establish the standard data representation for each step using XML.
  - Wrap software to provide data transformations as needed.

- Use the Grid technology to manage multiple clusters/supercomputers located across a wide area network.
Why Computational Grids?

- Performance characteristics of every step varies
  - First-principle and phase-filed simulation are very compute-intensive, CALPHAD is light weight, etc
  - Need scheduling to manage different components properly across local clusters/supercomputers
  - Need distributed access to intermediate databases

- Commercial software packages cannot be exported
  - Modifying source code is almost impossible and there are licensing issues
  - We create a virtual context that can be migrated to manage remote executions and data transfer for multiple remote executions
  - We provide extensible encapsulation of these packages to dynamically create a computing “service” dynamically to match clients’ designs
Software System of MATCASE

- Client-Server model of web-portal to access the Grid
  - Web page works as an user interface
  - Server implemented by Java Servlet
  - Servlet executes remote jobs using Globus Toolkit
  - Output is obtained through a web page
Example: VASP
Example: VASP

The output will be translated to construct input file for CALPHAD
XML Representation

```xml
<?xml version="1.0"?>
<compounds>
  <compound>
    <compound_name>AL SI</compound_name>
    <parameters>
      <parameter>
        <name>G(LIQUID,AL,SI;0)</name>
        <lowlimit>298.15</lowlimit>
        <functions>
          <function>
            <gibbs>V1+V2*T;</gibbs>
            <highlimit>6000</highlimit>
          </function>
        </functions>
      </parameter>
    </parameters>
  </compound>
</compounds>
```

Header of Al-Si system

Gibbs energy function
Example: Thermocalc
Example: Thermocalc

This will be transferred to phase-field simulation.
Example question:

- Compute mechanical response (stress-strain) curves for Al-Cu (5%Cu) system with fcc and q’ (metastable) phases at 280K-300K

- First principle calculation
  - Give lattice parameters based on experimental/theoretical models and execute a number of simulations

- CALPHAD
  - Setup the input file for CALPHAD software
  - New simulation results may affect the other systems

- Phase-field simulation
  - Finding Gibbs free energy function for q’ phases to set up the computation

- Finite Element Analysis
  - Assigning boundary conditions and other parameters for each domain that represents different phases
  - Choose linear solver and preconditioning scheme

- Overall
  - Provide a viable composition of the individual steps
  - Determine the number of processors, etc.
  - Recover from errors numerical, rely on Globus fault tolerance, etc.
Towards multicomponent material design

**Issues:**

- Identifying data necessary for the simulation
- Providing a default form of inputs
- Translating output between computational steps
- Managing workflow of the simulation sequence
- **Analyzing intermediate results to provide meaningful simulations (i.e. avoid cascading bad simulation results, detect failures to converge, etc.)**
Interaction handler

Simulation handler 1

Simulation handler 2

Simulation handler 3

Simulation handler 4

Rules Database

Web server

Remote resources

Credent repository (Myproxy)
Three Part System

Build a reconfigurable web portal system with 3 main modules

- **Interaction handler**
  - Takes care of presentation

- **Analyzer**
  - Creates instances of interaction handlers and simulation handlers
  - Management of “rules”
  - Bridge between interaction handler and simulation handler for each client

- **Simulation handler**
  - Execution of remote tasks using Globus
  - Creating instances of the local “services” to process input/output between steps
  - Transfer input/output for client between the server and remote computers
Interaction Handler

- Establish initial problem specification and further interaction to define constraints, refine model, present and evaluate results

- The first page for client is a simple HTML form
- After the input for the first page, interaction handler creates JSP scripts for input and output for client specific simulation using a specification from the analyzer
- The JSP scripts are saved in conjunction with the corresponding XML simulation specification in the analyzer
Analyzer

Analyzer creates the specification of simulation

- The input from a client is used along with the specification in the database to create the exact simulation instance
  - This specification is the set of rules and workflow
- The specification is sent to interaction handler and simulation handler as Java-XML object
  - The results from simulation handler is interpreted to help generate JSP file in the interaction handler
Analyzer (Continued)

- The typical rules and heuristics used in the materials science community are stored in simulation database in XML format

  - Analyzer is the **expert system**: Save information across simulations to allow **rule mapping** to similar simulations
Simulation Handler

- Handles the sequence of the execution of remote tasks:
  - The workflow is handled by a finite state transition automata/table
  - Each remote execution is modularized as a “task” module.
  - Information of a user is managed by a “user” module.
    - “User” activates its own “task” module
  - Fault tolerance should be supported
- All remote tasks are executed by Java CoG kit (Java interface of the Globus)
- Handles input and output between tasks, and remote resources and server
Interaction handler 1

JSP

Interaction handler 2

JSP

Interaction handler 3

JSP

Interaction handler 4

JSP

Analyzer

Rules Database

Simulation handler 1

Web server

Simulation handler 2

Simulation handler 3

Simulation handler 4

Remote resources

Credential repository (Myproxy)
Rule Based System

- Supports the construction of simulation workflow and fault detection (of simulation results)
- “Rules” define how to compose results of intermediate steps, etc
- The user interacts with a static web page
  - The input sent to the analyzer
  - The analyzer generates a specification of the simulation
  - The specification creates interaction and simulation handlers
  - Users, then, run the simulations with new interface
Service Creation

Client (Web browser) → Web Server

- Static webpage
- JSP for input
- JSP for output
- Analyzer
- Rule database
- Interaction handler
- Simulation handler
Execution of the Simulation

Client (Web browser) → Web Server → Remote computer

- JSP for input
- JSP for output
- Analyzer
- Interaction handler
- Simulation handler
Intermediate Tasks

- Translates data between successive computational steps using XML
- Generates the new C/Fortran code (such that new Gibbs energy function) and links it with the next program
- Tasks are managed by factory that spawns various local services dynamically

Diagram:

- Simulation handler
- task2
- Intermediate task
- task3
- CALPHAD → XML output
- Intermediate service (extract local rules)
- XML input
- Phase-field simulation
- Fortran subroutine
Conclusions

- We present our plan of the implementing our grid-enabled multicomponent materials modeling system:
  - Current status: only sequential modules are connected for single-line like workflows
  - Next: adding in parallel phase-field, new simulation hand to support complicated workflows
  - Later: studies of Grid performance
- The system is a client-server model with a backend that accesses the Grid
- Tools for debugging Globus programs and repository of any Grid-enabled applications will help!
Related Work

- Astrophysics Simulation Collaboratory (Max Planck Institute)
- Active Thermochemical Table Framework (ANL)
- GridSpeed (Tokyo Institute of Technology)
- XCAT (Indiana U.)
- Globus Toolkit 2.4, 3.2 and OGSA