Autonomic Grid Computing: Concepts, Infrastructure and Applications

Manish Parashar
The Applied Software Systems Laboratory
ECE/CAIP, Rutgers University

http://www.caip.rutgers.edu/TASSL

(Ack: NSF, DoE, NIH)
Autonomic Grid Computing: Concepts, Infrastructure and Applications

Abstract: Emerging pervasive wide-area Grid computing environments are enabling a new generation of applications that are based on seamless aggregation and interactions of resources, services and information. However the scale, dynamism and uncertainty of these environments and applications present significant development, configuration and management challenges. Addressing these challenges has led researchers to consider alternative programming paradigms and management techniques that are based on strategies used by biological systems to deal with complexity, dynamism, heterogeneity and uncertainty. The approach, referred to as autonomic computing, aims at realizing computing systems and applications capable of managing themselves with minimal human intervention. In this talk I will motivate and introduce autonomic Grid computing. I will then introduce solutions being developed at TASSL, Rutgers University as part of Project AutoMate for enabling autonomic computational science on the Grid.
Outline

• Pervasive Grid Environments - Unprecedented Opportunities

• Pervasive Grid Environments - Unprecedented Challenges

• Autonomic Grid Computing

• Project AutoMate @ TASSL, Rutgers University – Enabling Autonomic Applications in Pervasive Grid Environments

• An Illustrative Application

• Concluding Remarks
The original Grid concept has moved on!

- Coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations.

Source: I. Foster et al
Emerging Information Infrastructures - Smaller/Cheaper/Faster/Powerful/Connected ….

• Explosive growth in computation, communication, information and integration technologies
  – *Computing & communication is ubiquitous*

• Pervasive ad hoc “anytime-anywhere” access environments
  – *Ubiquitous access to information*
  – *Peers capable of producing/consuming/processing information at different levels and granularities*
  – *Embedded devices in clothes, phones, cars, mile-markers, traffic lights, lamp posts, medical instruments …*

• “On demand” computational/storage resources, services
Pervasive Grid Environments and Information Driven Applications

Experts query, configure resources

Experts interact and collaborate using ubiquitous and pervasive portals

Resources discovered, negotiated, co-allocated on-the-fly. Components deployed

Experts monitor/interact with/interrogate/steer models (“what if” scenarios,…). Application notifies experts of interesting events.

Components dynamically composed. “WebServices” discovered & invoked.

Real-time data assimilation/injection (sensors, instruments, experiments, data archives).

Data Archive & Sensors

Automation mining & matching

Computers, Storage, Instruments, ...

Applications & Services

Experts mine archive, match real-time data with history

Data Archives

Sensors, Non-Traditional Data Sources

Model A

Model B

Scientist

Scientist

Laptop

PDA

Computer
Pervasive Grid Environments - Unprecedented Opportunities

• **Pervasive Grids Environments**
  – *Seamless, secure, on-demand access to and aggregation of, geographically distributed computing, communication and information resources*
    • Computers, networks, data archives, instruments, observatories, experiments, sensors/actuators, ambient information, etc.
  – *Context, content, capability, capacity awareness*
  – *Ubiquity and mobility*

• **Knowledge-based, information/data-driven, context/content-aware computationally intensive, pervasive applications**
  – *Symbiotically and opportunistically combine services/computations, real-time information, experiments, observations, and to manage, control, predict, adapt, optimize, …*
    • Crisis management, monitor and predict natural phenomenon, monitor and manage engineering systems, optimize business processes

• **A new paradigm ?**
  – *seamless access*
    • resources, services, data, information, expertise, …
  – *seamless aggregation*
  – *seamless (opportunistic) interactions/couplings*
Sun Proposal: Integration of Sensors and High End Computers using Network

The Network Computer

Core

Global Network

Devices

Things

Data Center

Service Edge

Access Edge

Platform

Sun Remote Services

Source: Anil Velluri, Sun
Information-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.
Vision: Diverse Geosystems – Similar Solutions

Landfills

Oilfields

Underground Pollution

Undersea Reservoirs

Models

Simulation

Control

Data
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
  – Flowmeter at bottom of dump
  – Weather-station
  – Manually sampled chemical/air ports in wells
  – Approx 40K measurements/day

• 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.

Many Application Areas ....

- Hazard prevention, mitigation and response
  - Earthquakes, hurricanes, tornados, wild fires, floods, landslides, tsunamis, terrorist attacks
- Critical infrastructure systems
  - Condition monitoring and prediction of future capability
- Transportation of humans and goods
  - Safe, speedy, and cost effective transportation networks and vehicles (air, ground, space)
- Energy and environment
  - Safe and efficient power grids, safe and efficient operation of regional collections of buildings
- Health
  - Reliable and cost effective health care systems with improved outcomes
- Enterprise-wide decision making
  - Coordination of dynamic distributed decisions for supply chains under uncertainty
- Next generation communication systems
  - Reliable wireless networks for homes and businesses


Source: M. Rotea, NSF
Outline

• Pervasive Grid Environments - Unprecedented Opportunities

• Pervasive Grid Environments - Unprecedented Challenges
  – System, Information, Application Uncertainty

• Autonomic Grid Computing

• Project AutoMate @ TASSL, Rutgers University – Enabling Autonomic Applications in Pervasive Grid Environments

• An Illustrative Application

• Concluding Remarks
Pervasive Grid Applications – Unprecedented Challenges: Uncertainty

• System Uncertainty
  – Very large scales
  – Ad hoc structures/behaviors
    • p2p, hierarchical, etc, architectures
  – Dynamic
    • entities join, leave, move, change behavior
  – Heterogeneous
    • capability, connectivity, reliability, guarantees, QoS
  – Lack of guarantees
    • components, communication
  – Lack of common/complete knowledge
    • number, type, location, availability, connectivity, protocols, semantics, etc.

• Information Uncertainty
  – Availability, resolution, quality of information
  – Devices capability, operation, calibration
  – Trust in data, data models
  – Semantics

• Application Uncertainty
  – Dynamic behaviors
    • space-time adaptivity
  – Dynamic and complex couplings
    • multi-physics, multi-model, multi-resolution, ....
  – Dynamic and complex (ad hoc, opportunistic) interactions
  – Software/systems engineering issues
    • Emergent rather than by design
Pervasive Grid Computing – Research Issues, Opportunities

• Programming systems/models for data integration and runtime self-management
  – components and compositions capable of adapting behavior, interactions and information
  – correctness, consistency, performance, quality-of-service constraints

• Content-based asynchronous and decentralized discovery and access services
  – semantics, metadata definition, indexing, querying, notification

• Data management mechanisms for data acquisition and transport with real time, space and data quality constraints
  – high data volumes/rates, heterogeneous data qualities, sources
  – in-network aggregation, integration, assimilation, caching

• Runtime execution services that guarantee correct, reliable execution with predictable and controllable response time
  – data assimilation, injection, adaptation

• Security, trust, access control, data provenance, audit trails, accounting
Outline

• Pervasive Grid Environments - Unprecedented Opportunities

• Pervasive Grid Environments - Unprecedented Challenges

• Autonomic Grid Computing

• Project AutoMate @ TASSL, Rutgers University – Enabling Autonomic Applications in Pervasive Grid Environments

• An Illustrative Application

• Concluding Remarks
Integrating Biology and Information Technology: The Autonomic Computing Metaphor

• Current programming paradigms, methods, management tools are inadequate to handle the scale, complexity, dynamism and heterogeneity of emerging systems.

• Nature has evolved to cope with scale, complexity, heterogeneity, dynamism and unpredictability, lack of guarantees:
  - self configuring, self adapting, self optimizing, self healing, self protecting, highly decentralized, heterogeneous architectures that work!!!

• Goal of autonomic computing is to build a self-managing system that addresses these challenges using high level guidance.

Adaptive Biological Systems

• The body’s internal mechanisms continuously work together to maintain essential variables within physiological limits that define the viability zone.

• Two important observations:
  – *The goal of the adaptive behavior is directly linked with the survivability.*
  – *If the external or internal environment pushes the system outside its physiological equilibrium state the system will always work towards coming back to the original equilibrium state.*
Ashby’s Ultrastable System
Autonomic Computing Characteristics (IBM)

Self-protecting System designed to protect itself from any unauthorized access anywhere.

Self-optimizing System designed to automatically manage resources to allow the servers to meet the enterprise needs in the most efficient fashion.

Self-configuring systems designed to define itself "on the fly".

Self-healing Autonomic problem determination and resolution.
Autonomic Computing Architecture

- Autonomic elements (components/services)
  - Responsible for policy-driven self-management of individual components

- Relationships among autonomic elements
  - Based on agreements established/maintained by autonomic elements
  - Governed by policies
  - Give rise to resiliency, robustness, self-management of system
Autonomic Elements: Structure

- Fundamental atom of the architecture
  - Managed element(s)
    - Database, storage system, server, software app, etc.
  - Plus one autonomic manager

- Responsible for:
  - Providing its service
  - Managing its own behavior in accordance with policies
  - Interacting with other autonomic elements

Ack. IBM
Autonomic Elements: Interactions

• Relationships
  – Dynamic, ephemeral, opportunistic
  – Defined by rules and constraints
  – Formed by agreement
    • May be negotiated
  – Full spectrum
    • Peer-to-peer
    • Hierarchical
  – Subject to policies
Autonomic Elements: Composition of Autonomic Systems

Ack. IBM
Autonomic Computing: Research Issues and Challenges

Scale, Complexity, Dynamism, Heterogeneity, Unreliability, Uncertainty

- Defining autonomic elements
  - Programming paradigms and development models/frameworks
    - Autonomic element definition and construction
    - Rule definition, representation, and enforcement
- Constructing autonomic systems/applications
  - Composition, coordination, interactions models and infrastructures
    - Dynamic (rule-based) configuration, execution and optimization
    - Dynamic (opportunistic) interactions, coordination, negotiation
- Execution and management
  - Runtime and middleware services
    - Discovery, Coordination, Messaging, Security, Management, ...
  - Security, protection
  - Fault tolerance, reliability, availability, ...
- Policies, Learning, AI, ...
Outline

• Pervasive Grid Environments - Unprecedented Opportunities

• Pervasive Grid Environments - Unprecedented Challenges, Opportunities

• Autonomic Grid Computing

• Project AutoMate @ TASSL, Rutgers University – Enabling Autonomic Applications in Pervasive Grid Environments

• An Illustrative Application

• Concluding Remarks
Programming Pervasive Grid Systems

– Programming System
  • programming model, languages/abstraction – syntax + semantics
    – entities, operations, rules of composition, models of coordination/communication
  • abstract machine, execution context and assumptions
  • infrastructure, middleware and runtime

– Hide or expose uncertainty?
  • robustness, ease of programming
  • the inverted stack ...

Autonomic Grid Computing – A Holistic Approach

• Computing has evolved and matured to provide specialized solutions to satisfy relatively narrow and well defined requirements in isolation
  – performance, security, dependability, reliability, availability, throughput, pervasive/amorphous, automation, reasoning, etc.

• In case of pervasive Grid applications/environments, requirements, objectives, execution contexts are dynamic and not known a priori
  – requirements, objectives and choice of specific solutions (algorithms, behaviors, interactions, etc.) depend on runtime state, context, and content
  – applications should be aware of changing requirements and executions contexts and to respond to these changes are runtime

• Autonomic Grid computing - systems/applications that self-manage
  – use appropriate solutions based on current state/context/content and based on specified policies
  – address uncertainty at multiple levels
  – asynchronous algorithms, decoupled interactions/coordination, content-based substrates
Project AutoMate: Enabling Autonomic Applications

• Conceptual models and implementation architectures
  – *programming systems based on popular programming models*
    • *object, component and service based prototypes*
  – *content-based coordination and messaging middleware*
  – *amorphous and emergent overlays*
• [http://automate.rutgers.edu](http://automate.rutgers.edu)
Project AutoMate: Core Components

- Accord – A Programming System for Autonomic Grid Applications
- Squid – Decentralized Information Discovery and Content-based Routing
- Meteor – Content-based Interactions/Messaging Middleware
- Rudder/Comet – Decentralized Coordination Middleware
- ACE – Autonomic Composition Engine
- SESAME – Context-Aware Access Management
- DAIS – Cooperative Protection against Network Attacks

More information/Papers – [http://automate.rutgers.edu](http://automate.rutgers.edu)

Accord: Rule-Based Programming System

Accord is a programming system which supports the development of autonomic applications.

- Enables definition of autonomic components with programmable behaviors and interactions.
- Enables runtime composition and autonomic management of these components using dynamically defined rules.
  - Dynamic specification of adaptation behaviors using rules.
  - Enforcement of adaptation behaviors by invoking sensors and actuators.
  - Runtime conflict detection and resolution.

3 Prototypes: Object-based, Components-based (CCA), Service-based (Web Service)

Autonomic Element/Behaviors in Accord

Element Manager

Computational Element
  Functional Port
  Control Port
  Operational Port

Autonomic Element

Application workflow

Application strategies
  Application requirements

Composition manager

Interaction rules

Behavior rules

Interaction rules

Behavior rules

Interaction rules

Behavior rules

Event generation

Other Interface invocation

Actuator invocation

Internal state

Rules

Contextual state
LLC Based Self Management within Accord

- Element/Service Managers are augmented with LLC Controllers
  - monitors state/execution context of elements
  - enforces adaptation actions determined by the controller
  - augment human defined rules
The Self-managing Shock Simulation: Self-optimizing Via Component Replacement

1. register cache miss event
2. collect cache miss of GodunovFlux
3. evaluate the rule
   IF cache miss of GodunovFlux > value
   THEN REPLACE GodunovFlux
   EFMFlux
4. replace GodunovFlux with EFMFlux

EFMFlux will be used from the next computation
The Self-managing Shock Simulation: Self-optimizing Via Component Adaptation

1. export actuator "algorithm"
2. register communication bandwidth
3. collect current bandwidth
4. evaluate the rule
   IF bandwidth < threshold
   THEN algorithm x
5. invoke algorithm with x

Component Manager

Performance toolkit (TAU)

AMRMesh

Algorithm x will be used from the next computation
The Self-managing Shock Simulation: Self-healing Via Component Replacement

1. register execution error as a sensor

2. evaluate the rule

   IF GodunovFlux error
   THEN REPLACE GodunovFlux EFMFlux

3. replace GodunovFlux with EFMFlux
Decentralized (Decoupled/Asynchronous) Content-based Middleware Services

Middleware Services

- Discovery (Squid)
- Communication (Meteor)
- Coordination (Comet)

Content-Overlay mapping / routing (Squid)

Self-Managing Overlay (SquidTON)

Pervasive Grid Environment
SquidTON: Reliable & Fault Tolerant Overlay

- Pervasive Grid systems are dynamic, with nodes joining, leaving and failing relatively often
  - => *data loss and temporarily inconsistent overlay structure*
  - => *the system cannot offer guarantees*
    - *Build redundancy into the overlay network*
    - *Replicate the data*

- SquidTON = Squid Two-tier Overlay Network
  - *Consecutive nodes form unstructured groups, and at the same time are connected by a global structured overlay (e.g. Chord)*
  - *Data is replicated in the group*
Content Descriptors and Information Space

- Data element = a piece of information that is indexed and discovered
  - Data, documents, resources, services, metadata, messages, events, etc.
- Each data element has a set of keywords associated with it, which describe its content => data elements form a keyword space

2D keyword space for a P2P file sharing system

3D keyword space for resource sharing, using the attributes: storage space, base bandwidth and cost
Content Indexing: Hilbert SFC

- $f: \mathbb{N}^d \rightarrow \mathbb{N}$, recursive generation

**Properties:**
- Digital causality
- Locality preserving
- Clustering

*Cluster:* group of cells connected by a segment of the curve
Content Indexing, Routing & Querying

Content profile 2: (4-7,0-3)
Content profile 3: (*, 4)

- Demonstrated analytically and experimentally that
  - for large systems that queries with p% coverage will query p% of the nodes, independent on data distribution.
  - the system scales with the number of nodes and data
  - optimization significantly reduces the number of clusters generated and messages sent
  - Slightly increases the number of nodes queried – only a small number of “intermediary” nodes are involved

- Note:
  - More than one cluster are typically stored at a node
  - Not all clusters that are generated for a query exist in the network
  - SFC, clusters generation is recursive, i.e., a prefix tree (trie)
- Optimization: embed the tree into the overlay and prune nodes during construction
Project Meteor: Associative Rendezvous

- Content-based decoupled interaction with programmable reactive behaviors
  - Messages - (header, action, data)
    - Symmetric post primitive: does not differentiating between interest/data
  - Associative selection
    - match between interest and data
  - Reactive behavior
    - Execute action field upon matching
- Decentralized in-network aggregation
  - Tries for back-propagating and aggregating matching data items
- Supports WS Notification standard

Profile = list of (attribute, value) pairs:
Example:
\(<(sensor\_type, temperature), (latitude, 10), (longitude, 20)>\)
Heterogeneity Management

- Heterogeneity management and adaptations at AR nodes using reactive behavior
  - Policy-based adaptations based on capabilities, preferences, resources
Implementation/Deployment Overview

- Current implementation builds on JXTA
  - SquidTON, Squid, Comet and Meteor layers are implemented as event-driven JXTA services

- Deployments include
  - Campus Grid @ Rutgers
  - Orbit wireless testbed (400 nodes)
  - PlanetLab wide-area testbed
    - At least one node selected from each continent
Outline

• Pervasive Grid Environments - Unprecedented Opportunities

• Pervasive Grid Environments - Unprecedented Challenges, Opportunities

• Autonomic Grid Computing

• Project AutoMate @ TASSL, Rutgers University – Enabling Autonomic Applications in Pervasive Grid Environments

• An Illustrative Application

• Concluding Remarks
Wide Area Data Streaming in the Fusion Simulation Project (FSP)

- Wide Area Data Streaming Requirements
  - Enable high-throughput, low latency data transfer to support near real-time access to the data
  - Minimize related overhead on the executing simulation
  - Adapt to network conditions to maintain desired QoS
  - Handle network failures while eliminating data loss.
Autonomic Data Streaming for Coupled Fusion Simulation Workflows

- Grid-based coupled fusion simulation workflow
  - Run for days between ORNL (TN), NERSC (CA), PPPL (NJ) and Rutgers (NJ)
  - Generates multi-terabytes of data
  - Data coupled, analyzed and visualized at runtime
ADSS Implementation

- Data Rate Prediction
- Buffer size Prediction
- LLC Model
  - Optimization Function
  - Service State
  - Contextual State
- Data Rate Measurement
- Bandwidth Measurement
- Buffer Size Measurement
- Rule Base
  - BMS
  - DTS
  - ADSS
  - Grid Middleware {LN}
  - Backbone
  - Local HD

Rule Based Adaptation

LLC Controller
Element/Service Manager
Adaptive Data Transfer

- No congestion in intervals 1-9
  - Data transferred over WAN
- Congested at intervals 9-19
  - Controller recognizes this congestion and advises the “Element Manager” which in turn adapts DTS to transfer data to local storage (LAN).
- Adaptation continues until the network is not congested
  - Data sent to the local storage by the DTS falls to zero at the 19th controller interval.
Adaptive Buffer Management

- Uniform buffer management is used when data generation is constant
- Aggregate buffer management is triggered when congestion increases
Adaptation of the Workflow

- Create multiple instances of the Autonomic Data Streaming Service (ADSS)
  - Effective Network Transfer Rate dips below the Threshold (our case around 100Mbs)
  - Maximum number of instances is contained within a certain limit

% Network throughput is difference between the max and current network transfer rate
The Instrumented Oil Field of the Future (UT-CSM, UT-IG, RU, OSU, UMD, ANL)

- Production of oil and gas can take advantage of installed sensors that will monitor the reservoir’s state as fluids are extracted.
- Knowledge of the reservoir’s state during production can result in better engineering decisions:
  - economical evaluation; physical characteristics (bypassed oil, high pressure zones); productions techniques for safe operating conditions in complex and difficult areas.

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, future data acquisition strategy

Effective Oil Reservoir Management: Well Placement/Configuration

• Why is it important
  – **Better utilization/cost-effectiveness of existing reservoirs**
  – **Minimizing adverse effects to the environment**

![Bad Management](image1.png)
**Much Bypassed Oil**

![Better Management](image2.png)
**Less Bypassed Oil**
An Autonomic Well Placement/Configuration Workflow

If guess not in DB:
- Instantiate IPARS with guess as parameter
- Send guesses to MySQL Database
- Optimization Service
- Generate Guesses

If guess in DB:
- Send response to Clients
- Get new guess from Optimizer
- IPARS Factory
- Start Parallel IPARS Instances
- Instance connects to DISCOVER
- DISCOVER
- Notifies Clients
- Clients interact with IPARS

AutoMate Programming System/Grid Middleware

- History/Archived Data
- Sensor/Context Data
- Oil prices, Weather, etc.
Autonomic Oil Well Placement/Configuration

Permeability

Pressure contours
3 wells, 2D profile

Contours of NEval(y,z,500)(10)

Requires NYxNZ (450) evaluations. Minimum appears here.

VFSA solution: “walk”:
found after 20 (81) evaluations
Autonomic Oil Well Placement/Configuration (VFSA)


Outline

• Pervasive Grid Environments - Unprecedented Opportunities

• Pervasive Grid Environments - Unprecedented Challenges, Opportunities

• Autonomic Grid Computing

• Project AutoMate @ TASSL, Rutgers University – Enabling Autonomic Applications in Pervasive Grid Environments

• An Illustrative Application

• Concluding Remarks
• Pervasive Grid Environments - Unprecedented Opportunities

• Pervasive Grid Environments - Unprecedented Challenges, Opportunities

• Project AutoMate @ TASSL, Rutgers University – Enabling Autonomic Applications in Pervasive Grid Environments

• An Illustrative Application

• Concluding Remarks
Conclusion

• Pervasive Grid Environments
  – Unprecedented opportunity
    • can enable a new generation of knowledge-based, data and information driven, context-aware, computationally intensive, pervasive applications
  – Unprecedented research challenges
    • scale, complexity, heterogeneity, dynamism, reliability, uncertainty, …
    • applications, algorithms, measurements, data/information, software

• Autonomic Grid Computing
  – Addressing the complexity of pervasive Grid environments

• Project AutoMate: Autonomic Computational Science on the Grid
  – Semantic + Autonomics
  – Accord, Rudder/Comet, Meteor, Squid, Topos, …

• More Information, publications, software
  – www.caip.rutgers.edu/~parashar/
  – parashar@caip.rutgers.edu
The Team

- **TASSL, CAIP/ECE Rutgers University**
  - Viraj Bhat
  - Sumir Chandra
  - Andres Q. Hernandez
  - Nanyan Jiang
  - Zhen Li (Jenny)
  - Vincent Matossian
  - Cristina Schmidt
  - 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. Schwan, M. Wolf
  - University of Maryland
    - A. Sussman, C. Hansen

- **Key Applications Collaborators**
  - Rutgers Univ.
    - R. Levy, S. Garofilini
  - 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
    - T. –C. J. Yeh, J. Daniels, A. Kruger
  - Idaho National Laboratory
    - R. Versteeg
  - PPPL
    - R. Samtaney
  - ASCI/CACR, Caltech
    - J. Cummings
Thank You!

There are no stupid questions, just stupid answers ...
Fire at will!