Autonomic Oil Reservoir Optimization using Decentralized Services

Manish Parashar, The AutoMate Group
The Applied Software Systems Laboratory
Rutgers, The State University of New Jersey
http://automate.rutgers.edu
Ack: NSF (CAREER, KDI, ITR, NGS), DoE (ASCI, CIT)

SUN HPC
November 16, 2003
Outline

• Autonomic Grid Computing – Enabling Seamless Interactions
• Autonomic Interactions for Oil Reservoir Optimization
• AutoMate: Enabling Autonomic Applications
  – Pawn: P2P Messaging Substrate for Autonomic Interactions
• Conclusions
Grid Computing: Seamless Aggregation

“Resource sharing & coordinated problem solving in dynamic, multi-institutional virtual organizations”
Autonomic Grid Computing: Seamless Interaction for Global Scientific Investigation

- Secure/Seamless Interactions
  - Client – Client -> Collaboration,
  - Client – Application -> Monitoring/Interaction/Steering,
  - Application – Application -> Application components, services,
  - Application – Data -> Dynamic Data Injection, Sensor networks, I/O, Checkpoint/Restart,
  - Client – Data -> Visualization, mining,
  - Application – Resource -> Staging, dynamic resource allocation, execution migration

- Autonomic Infrastructure
  - Autonomous components and dynamic composition, configuration, optimizations
    - Self defining, self self-defining, self-configuring, self-optimizing, self-protecting, self-healing, context aware and anticipatory
    - “Deductive” computational middleware
      - Dynamic, rule-based, component configuration, optimization, and composition, constraint-based dynamic interactions
  - P2P Grid infrastructure
    - Security, dynamic access control, discovery, messaging, resource management, QoS, collaboration, interaction,
AutoMate: Enabling Autonomic Applications
(http://automate.rutgers.edu)

- **Objective:**
  - To enable the development of autonomic Grid applications that are context aware and are capable of self-configuring, self-composing, self-optimizing and self-adapting.
  - address uncertainty, unreliability, dynamism, incomplete knowledge

- **Overview:**
  - **Definition of Autonomic Components**
    - definition of programming abstractions and supporting infrastructure that will enable the definition of autonomic components
    - autonomic components provide enhanced profiles or contracts that encapsulate their functional, operational, and control aspects
  
  - **Dynamic Composition and Coordination of Autonomic Applications**
    - models, mechanisms and supporting infrastructure to enable autonomic applications to be dynamically and opportunistically composed from autonomic components
    - composition/coordination based on (dynamically defined) declarative policies, goals and constraints, and aware of available Grid resources (systems, services, storage, data) and components, and their current states, requirements, and capabilities

  - **Autonomic Middleware Services**
    - design, development, and deployment of key services on top of the Grid middleware infrastructure to support autonomic applications
    - enable components, applications and resources (systems, services, storage, data) to interact as peers
AutoMate: Architecture

• Key components:
  – Accord: Autonomic application framework
  – Rudder: Decentralized deductive engine
  – Squid: P2P discovery service
  – SESAME: Dynamic access control engine
  – Pawn: P2P messaging substrate
A Data Intense Challenge: The Instrumented Oil Field of the Future (UT-CSM, UT-IG, RU, OSU, UMD, ANL)

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

- Production simulation via reservoir modeling
- Monitor production by acquiring time lapse observation of seismic data
- Revise knowledge of reservoir model via imaging and inversion of seismic data
- Modify production strategy using an optimization criteria
Autonomic Oil Well Placement (UT-CSM, UT-IG)

- Optimization algorithm: use VFSA (Very Fast Simulated Annealing)
  - requires function evaluation only, no gradients
- IPARS delivers
  - fast-forward model (guess->objective function value)
  - post-processing
- Formulate a parameter space
  - well position and pressure \((y,z,P)\)
- Formulate an objective function:
  - maximize economic value \(Eval(y,z,P)(T)\)
- Normalize the objective function \(NEval(y,z,P)\) so that:

\[
\min NEval(y,z,P) \iff \max Eval(y,z,P)
\]
Components of the AORO Application

- **IPARS : Integrated Parallel Accurate Reservoir Simulator**
  - *Parallel reservoir simulation framework*

- **IPARS Factory**
  - *Configures instances of IPARS simulations*
  - *Deploys them on resources on the Grid*
  - *Manages their execution*

- **VFSA : Very Fast Simulated Annealing**
  - *Optimizes the placement of wells and the inputs (pressure, temperature) to IPARS simulations.*

- **Economic Modeling Service**
  - *Uses IPARS simulations outputs and current market parameters (oil prices, costs, etc.) to compute estimated revenues for a particular reservoir configuration.*

- **DISCOVER Computational Collaboratory**
  - *Interaction & Collaboration*
  - *Distributed Interactive Object Substrate (DIOS)*
  - *Collaborative Portals*
IPARS Factory discovers and initializes VFSA Optimization Service.

Client configures and launches IPARS Factory and VFSA Optimization peers on resource of choice.

IPARS Factory gets initial guess from VFSA Optimization Service launches IPARS instance on resource of choice.

One optimal well placement is determined, IPARS Factory launches IPARS run.

Current oil price, market state, etc.

Scientists/Engineers collaboratively interact with IPARS.

Client can configure IPARS params.

IPARS connects to VFSA Optimization Services and presents revenue.

IPARS Factory generates new well placement.

Current oil price, market state, etc.
Pawn: Conceptual Overview

“Peers compose messages handled by services through specific interaction modalities”

Architecture of the Optimization Application components

Interactions
Synchronous/Asynchronous; Dynamic Data Injection; Remote Procedure Calls

Services
Application Execution; Application Runtime Control; Application Monitoring and Steering; Collaboration

Messages
Platform-independent; Coordination; Guarantees

Peers
Client; Rendezvous; Application

IPARS Factory  VFSA Optimization  DISCOVER Collaboratory  Economic Model

Pawn

GRID / JXTA
Pawn : Functionalities Overview

- Provides messaging mechanisms to enable interactions on the computing Grid
  - *publish/subscribe architecture across peer-to-peer domains*
- Builds high-level messaging semantics on top of low-level interaction modalities:
  - *PUSH*: e.g. dynamic data injection
  - *PULL*: e.g. monitoring
  - *REQUEST/RESPONSE*: e.g. data interrogation
  - *TRANSACTION*: e.g. steering
  - *FILTERED MULTICAST*: e.g. group collaboration
- Builds on Project JXTA
Project JXTA: Protocols

- **PDP** (Peer discovery protocol) : used by peers to advertise their own resources
- **PIP** (Peer Information protocol) : monitoring peers status and load
- **PBP** (Pipe Binding Protocol) : to establish a virtual communication channel between peers
- **PRP** (Peer Resolver Protocol) : sending and receiving queries and responses
- **RVP** (Rendezvous Protocol) : to propagate messages in a peer group
- **ERP** (Endpoint Routing Protocol) : to find routes from a source to a destination
From JXTA to Pawn

- **JXTA Provides core capabilities**
  - *Publication:*
    - endpoints publish uniquely identified messages.
  - *Advertisement:*
    - language-independent document describing a resource
  - *Caching :*
    - RV peers cache advertisements made by every endpoint and maintain consistent replicas.
  - *Routing :*
    - path to destination is determined by the nearest rendezvous peer using the endpoint router protocol

- **Pawn extends JXTA to provide**
  - *Distributed object Interaction on top of a peer-to-peer substrate*
    - marshaling/un-marshaling of Objects to XML streams
    - remote Procedure Calls (PawnRPC) using XML messages
  - *Interest Subscription*
    - content-based information dissemination. Every message contains metadata that allow peers to register interest on an attribute basis.
    - filtering at source or at destination
Pawn Interactions (1)

- JXTA communication enabled through:
  - **Pipe**
    - *asynchronous: non-blocking*
  - **Resolver**
    - *end-to-end messaging*
      - **TCP Stream**
      - **Datagram packets**
    - **broadcast/multicast**

- Pawn services build on JXTA communication mechanisms
Pawn Interactions (2)

- **Stateful messages**
  - *Every message is self-describing and self-sufficient*
  - *Can be resent by intermediary peers in case of failure*

- **Message Guarantees**
  - *Using FIFO queues and timeouts on a per message basis*

- **Synchronous/Asynchronous Communication**
  - *Blocking on pipe input channel until response (for synch)*
  - *Using Non-Blocking pipes and resolve service (for asynch)*

- **Dynamic Data Injection**
  - *Using event channels for dynamic data input*

- **Remote Procedure Calls (PawnRPC)**
  - *Synch/asynch RPC using XML for invoking remote methods and provide seamless application deployment across loosely-coupled domains*
Pawn Services

- Application Execution [AEX]
  - Start, stop and get status of Applications
- Application Monitoring and Steering [AMS]
  - Application querying and management
- Application Runtime and Control [ARC]
  - Publishes application responses and status
- Group communication
  - Handles text messages between groups of clients
Scenario: Peer Deployment

- Client authenticates to the DISCOVER Server running Globus toolkit using GSI
- Once authenticated, Clients can deploy IPARS Factory and VFSA optimization peers using Globus GRAM protocol on available machines
Scenario: Peer Discovery

- Peers publish advertisements describing their identity and functionalities.
- Using underlying JXTA Discovery services, peers discover the advertisements and can start interacting.
VFSA sends a well position guess to IPARS Factory

IPARS Factory checks in Database if guess has already been run
  - If guess found, result is returned to the clients and a new guess from VFSA is generated
  - If not found an IPARS instance is run

IPARS returns the normalized revenue value to VFSA Optimization
Scenario: Production Run for Monitoring and Steering

- Experts use client portals to collaboratively connect to the running application for monitoring and steering
Autonomic Oil Well Placement

Permeability

Pressure contours
3 wells, 2D profile

Requires NYxNZ (450) evaluations. Minimum appears here.

VFSA solution: “walk”:
found after 20 (81) evaluations
Autonomic Oil Well Placement using PAWN
Conclusion

• Autonomic Grid computing – seamless interactions for global scientific investigation
• AutoMate addresses key issues to enable the development of autonomic Grid applications
  – ACCORD: Autonomic application framework
  – RUDDER: Decentralized deductive engine
  – SESAME: Dynamic access control engine
  – Pawn: P2P messaging substrate
  – SQUID: P2P discovery service
• Autonomic oil reservoir optimization using Pawn/AutoMate
• More Information, publications, software, conference
  – http://automate.rutgers.edu
  – automate@caip.rutgers.edu / parashar@caip.rutgers.edu
  – http://www.autonomic-conference.org
Web Links

- Project AutoMate
  - [http://automate.rutgers.edu](http://automate.rutgers.edu)
- Pawn’s web page
  - [http://www.caip.rutgers.edu/~vincentm/PAWN](http://www.caip.rutgers.edu/~vincentm/PAWN)
- Project JXTA
  - [http://www.jxta.org](http://www.jxta.org)
- O’Reilly p2p web site
- Brendon Wilson’s book on JXTA
  - [http://www.brendonwilson.com](http://www.brendonwilson.com)
The Team

• TASSL Rutgers University
  – Autonomic Computing Research Group
    • Viraj Bhat
    • Manish Agarwal
    • Nanyan Jiang
    • Hua Liu (Maria)
    • Zhen Li (Jenny)
    • Manish Mahajan
    • Vincent Matossian
    • Cristina Schmidt
    • Guangsen Zhang
  – Autonomic Applications Research Group
    • Sumir Chandra
    • Xiaolin Li
    • Dave Roberts
    • Taher Saif
    • Li Zhang

• CS Collaborators
  – HPDC, University of Arizona
    • Salim Hariri
  – Biomedical Informatics, The Ohio State University
    • Tahsin Kurc, Joel Saltz
  – CS, University of Maryland
    • Alan Sussman, Christian Hansen

• Applications Collaborators
  – CSM, University of Texas at Austin
    • Hector Klie, Mary Wheeler
  – IG, University of Texas at Austin
    • Mrinal Sen, Paul Stoffa
  – ASCI/CACR, Caltech
    • Michael Aivazis, Julian Cummings, Dan Meiron
  – CRL, Sandia National Laboratory, Livermore
    • Jaideep Ray, Johan Steensland