Autonomic Applications

Autonomic Forest Fire Management

Manish Parashar
The Applied Software Systems Laboratory
Rutgers, The State University of New Jersey
http://automate.rutgers.edu

Salim Hariri
High Performance Distributed Computing Laboratory
The University of Arizona
http://www.ece.arizona.edu/~hpdc

ICAC 2004 Autonomic Computing Tutorial
May 16, 2004

Autonomic Forest Fire Simulation

Predicts fire spread (the speed, direction and intensity of forest fire front) as the fire propagates, based on both dynamic and static environmental and vegetation conditions.
Autonomic Computing Tutorial, ICAC 2004

Georeferenced Distributed Database (GDDB)

Wildfire Autonomic Runtime Manager (WARM)

Dynamic Data Driven Wildfire Model

Analysis Objectives

Virtual Computation Unit

Autonomic Scheduling

Forest Fire Cell Space: Groups of VCUs

Forest Fire Cell Space decomposed into VCUs
VCU and VRU

- Computational workloads are divided into Virtual Computational Units (VCU)
- Based on the availability of computing resources and their access policies, VCU's are scheduled on Virtual Resource Units (VRU)
- Implementation of VCU and VRU
  - Interfaces:
    i. Autonomic Component (VCU) State to be monitored by AM (Autonomic Manager)
    ii. VRU requirements by VCU (Output of Planner)
    iii. Action Enforce Interface (Globus RSL here)

Forest Fire Cell Autonomic Component: VCU
Composition and Execution of Distributed Forest Fire Simulation

Step 1: Initial Partitioning into Natural Regions – Online Monitoring & Analysis
Step 2: Trigger Planning Module
Step 3: Partition the NRs into VCUs – Planning Engine
Step 4 and 5: Autonomic Scheduling of VCUs on VRUs – Autonomic Scheduler
Step 6: VRUs are mapped onto actual physical resources and triggered for execution
Step 7: Sense the state of the application and the resources at runtime - Online Monitoring
Step 8: Dynamic Partitioning of Cell Space
Forest Fire Cell Space: Dynamic Repartitioning (Step 8)

<table>
<thead>
<tr>
<th>Simulation Time</th>
<th>Fire Spread with time</th>
</tr>
</thead>
<tbody>
<tr>
<td>T = 0</td>
<td>Fire not yet started</td>
</tr>
<tr>
<td>T = N</td>
<td>Fire propagates along North-West Direction</td>
</tr>
<tr>
<td>T = 2N</td>
<td>Fire propagates further, Burned areas are left behind</td>
</tr>
</tbody>
</table>

Initial partitioning

Burning zone finer gridding

Burned zone coarser gridding

Forest Fire Cell Space: Dynamic Addition and Deletion of Cells
**VCU States**

2. Ready: waiting for an external event to happen e.g. hear from a neighbor VCU.
3. Running: using a physical resource.
5. Not-Recently-Used: resource de-allocation based on Not-Recently-Used Algorithm.

**Resource De-allocation based on Not-Recently-Used Algorithm**

- VCU1: Imminent
- VCU2: Not-Recently-Used

![Diagram](image_url)
Dynamic Cell Space Partitioning: Self-Optimization

1  
2  
1 2 N

East or West
Wind Directions

Dynamic Cell Space Re-Partitioning: Self-Optimization

CCact: migrate cells
CCact: remove couplings
CCstate: loadHigh
CCstate: paused
CCstate: reconfigure
CCstate: ready
CCstate: loadMid
CCact: select cells for migration
CCact: resume

VCU1 before re-partitioning

VCU1 after re-partitioning

VCU2
Autonomic Design of Nanomaterials

**INPUT: COMPOSITION/PROCESSING**

- **Atomistics**: Simulate non-equilibrium solidification process, crystallization, diffusion and growth
- **Nanoscale**: Model the evolution and interaction of topologically complex nanosize metastable structures and their effective behavior
- **Microscale**: Model the collective behavior of assemblies of nanostructured particles

**Optimization**: Optimize metastable nanocomposition based on atomistic, nanoscale and macroscale properties.

**Computational Infrastructure**: To develop autonomic computational infrastructures and runtime management techniques for scalable parallel/distributing computing, automated interaction and data exchange between scales, real time sensing and computational response, collaborative monitoring and steering.

**OUTPUT: OPTIMIZED METASTABLE NANOCOMPOSITES**

---

**Conclusion**

- Autonomic applications are necessary to address scale/complexity/heterogeneity/dynamism/reliability challenges
- 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
- Application scenarios
  - Autonomic optimization of oil reservoirs
  - Autonomic runtime management
- More Information, publications, software, conference
  - [http://automate.rutgers.edu](http://automate.rutgers.edu)
  - automate@caip.rutgers.edu / parashar@caip.rutgers.edu
  - [http://www.autonomic-conference.org](http://www.autonomic-conference.org)