Pathways into Large Parameter Search Spaces: Experiences with Molecular Hyperdynamics

Justin M. Wozniak, Santanu Chatterjee, Paul Brenner, Douglas Thain, Aaron Striegel and Jesus A. Izaguirre

11th September 2008

Current location: Argonne National Laboratory, USA
wozniak@mcs.anl.gov
Scientific Repositories

- Start with user data requirements

```plaintext
type = dna
temperature = 300
integrator = langevin
```
Scientific Repositories

- Cluster-based simulation storage infrastructure
Scientific Repositories

- The desktop grid reality - we just have a lot of computers
- GEMS: Grid-Enabled Molecular Simulation
The Cooperative Network

- Where can I put my data?
- How can I keep it available?
- How will I find it again?
Motivation

- User needs:
  - Large scale file space
  - File organization
  - Storage site management
  - Access for existing software
  - **Workflow structures for large scale projects**

- Available resources:
  - Large, uncontrolled storage network
  - A searchable, parameterized replica management system
Outline

I. Runtime repositories
   - Database abstractions
   - Basic utilities

II. Workflow-like structures
   - Application in hyperdynamics
   - Event-driven workflows
   - Scalable job submission
A Storage Control Layer

- Repository users expect a high level abstraction layer
- The controller enables user-level administration of a dynamic system
**Puts & Gets**

- Automatic replica placement and location

---

**GEMS API**

**USER**

**Existing API**

**REPOSITORY**

- **GEMSput**
  - name=sorin
  - app=sim
  - seed=149

- **GEMSmatch**
  - name=sorin
  - app=sim
  - --key

- **GEMSget** $KEY$

- Storage placement map
  - { wombat, helios }

---

**Examples:**

- F.1
- F.2

- $KEY(s)$
GUI for General Purpose Operations

● Parameterized intermediate workflow data is easily browsed.
• Are these files available?
Fault Prioritization

- Automatically detect, prioritize, and correct faults
Parameterized Storage Organization

- Data structures support application-specific tagging and searches
Parameterized Storage Organization

- File placement managed by user-supplied topology information
Parameterized Storage Organization

- Topology information may be tapped when placing jobs or accessing replicas

USER

CONTROLLER

WOMBAT

SC0

PURDUE
Computation Among Replicas

- Replica-aware computing framework
Replica System Methods

- Basic methods:
  - Simple puts & gets
  - Replica location
  - Replica access site evaluation

- Streaming methods:
  - Advanced disk space reservation
  - I/O setup
  - I/O translation
Replica System Examples

- Archive creation
  
  **obtain sink:**
  
  > GEMSreserve 100MB

  **pipe command output:**
  
  > tar c dir | sink

  **or use parrot:**
  
  > parrot tar cf sink dir

- Job I/O setup
  
  > GEMSrun
  
  --input INPUT /$KEY/file.1
  
  name=Justin job=3
  
  --output OUTPUT file.2
  
  --exec job INPUT OUTPUT

- Simple shell setup for streaming replica system I/O
Example application: Hyperdynamics

- Enhanced, user-steered molecular simulation technique
- Algorithm implementation enhanced by the *data sweep* abstraction
Parameterized intermediate workflow data is easily accessed, rendered, and used.
Parameterized Workflow

- $u, r$: Parameter sweep variables (e.g. user, random seed)

- Parameter sweep dependency over time: (checkpointing)

  \[ S(u, r, t) : \]
  \[ S(u, r, t - 1) \]

- Parameter dependency with branches:

  \[ S(u, r, t, \text{branch} = p) : \]
  \[ S(u, r, t - 1, \text{branch} = p) \]
  \[ \text{or} \ S(\text{id}entity = p, \text{time} = t - 1) \]
Notification Tools

- Workflow element script
  
  **wait for match:**
  > GEMSnotify user=sorin
  >  \( r=4 \ t=3 \)
  
  **generate new record:**
  > GEMSrun ...

- Simply parameterize and send to background...
  (how much space is in the background?)
Scalable Notification

- Bottlenecking procedure allows for progress while limiting consumption of system resources.
Hyperdynamics Results

- Results for small simulated system (400 Argons)

<table>
<thead>
<tr>
<th>Method</th>
<th>Total (hours)</th>
<th>Turnaround (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HYD-DEPTH</td>
<td>1.8</td>
<td>1.8</td>
</tr>
<tr>
<td>HYD-BREADTH</td>
<td>5800</td>
<td>1.4</td>
</tr>
<tr>
<td>HYD-EXPLORE</td>
<td>6.2</td>
<td>1.4</td>
</tr>
</tbody>
</table>
I/O Ratio

- I/O Ratio
Recap

I. Storage management for distributed repositories
   - Virtual clustering, puts & gets
   - Fault prioritization, control loop framework
   - Parameterized abstraction layer
   - Data access for computation

II. Workflow model based on parameterized objects
   - Hyperdynamics application
   - Workflow formulation
   - Notification tools
   - Scalability
Summary of Results

- A controller model can help users administrate *ad hoc* storage networks.
- Prioritized storage management can improve data durability.
- Replica management systems can be integrated within a computation infrastructure.
- Parameterized workflows can form a simple building block for distributed data operations.

- **GEMS** is open source:
  [http://sourceforge.net/projects/gems-nd](http://sourceforge.net/projects/gems-nd)
Acknowledgments

• Collaborators:
  - Paul Brenner
  - Santanu Chatterjee
  - Douglas Thain
  - Aaron Striegel
  - Jesus Izaguirre
• NSF DBI-0450067
Future Work

- Formalization of the parameter generation and arithmetic
- Real-world applications of *grid derivation*.
- Repository interoperability for hybrid systems.
- Performance analysis for more complex cases.
• Icon workshop