Optimisation of Component Based Applications within a Grid Environment

John Darlington, Steven Newhouse, Nathalie Furmento, Anthony Mayer, Stephen McGough

London e-Science Centre
Imperial College, London
icpc-sw@doc.ic.ac.uk

Outline

- E-Science and The Grid
- ICENI extensible framework
- Using Meta-data for Run-Time Optimisation
- Example: Linear Systems
- Further developments
E-Science

- Networked Resources: The Grid
- Utilisation of the Grid
- Enabling Domain Specialists

E-Science & The Grid: Challenges

- Complex Applications
  - Computationally Intensive
  - Very Large Data Sets
- Complex Resources
  - Dynamic
  - Unreliable
- Accessibility
  - Portals
<table>
<thead>
<tr>
<th>Information!</th>
</tr>
</thead>
<tbody>
<tr>
<td>• User:</td>
</tr>
<tr>
<td>– Who, what, when, where</td>
</tr>
<tr>
<td>• Resources:</td>
</tr>
<tr>
<td>– Availability, character, policy</td>
</tr>
<tr>
<td>• Application:</td>
</tr>
<tr>
<td>– Composition, behaviour, performance</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Software Abstraction: Components</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Implementation + Interface</td>
</tr>
<tr>
<td>• Abstraction: Expressability</td>
</tr>
<tr>
<td>– Connection Oriented Programming</td>
</tr>
<tr>
<td>– Mobility</td>
</tr>
<tr>
<td>• Encapsulation: Native Code</td>
</tr>
<tr>
<td>– High Performance</td>
</tr>
<tr>
<td>– Multiple Implementations</td>
</tr>
</tbody>
</table>
User Roles

- **End-User**
  - Customisation: types → instances
  - Composition: instances → applications
- **Scientist**
  - Develop component types
  - Annotate with meta-information
- **Developer**
  - Develop component implementations
  - Annotate with meta-information

ICENI

The Iceni, under Queen Boudicca, united the tribes of South-East England in a revolt against the occupying Roman forces in AD60.

- IC E-Science Networked Infrastructure
- Developed by Grid Middleware Group
- Collect and provide relevant Grid meta-data
- Use to define and develop higher-level services
Static Application Information: Component XML (CXML)

- Application as network of components:
  - Application Description Document
- Component Interfaces & Behaviour:
  - Repository
- Implementation Performance Characteristics:
  - Repository

Conventional Component Software Lifecycle
Augmented Component Software Lifecycle

Advantages of Very Late Binding

- Mobility of the high-level representation
- Performance of low-level implementation
- Dynamic Implementation Selection
- Cross-component Optimisation
Performance Guided Implementation Selection

1. Application Composition
   - From Run-Time Representation
2. Implementation Performance Models
   - From Run-Time Representation
3. Resource Information
   - From Grid Infrastructure
4. Selection Policy
   - From Component Framework

Example: Linear Solver
Decision Making (3)

<repository>
<component package="icpc.LinearEquationSolver" name="LinearSolverRowsColumnsUnsymmetric" version="1">
<parent package="icpc.framework.core" name="Component"/>
<inPort objectPackage="icpc.core" objectName="Integer" portName="unknowns"/>
<inPort objectPackage="icpc.Matrix" objectName="DgeRCj" portName="matrix"/>
<inPort objectPackage="icpc.Matrix" objectName="RealVector" portName="vector"/>
<data exposure="public" objectPackage="icpc.Matrix" objectName="RealVector" dataName="solution">
<outPort>
<behaviour method="getVector">
<block execute="sequential">
  <get name="m1" portName="matrix" method="getMatrix"/>
  <get name="dof" portName="unknowns" method="getInteger"/>
  <get name="v1" portName="vector" method="getVector"/>
  <call portName="solution" method="getVector"/>
</block>
</behaviour>
</outPort>
</data>
<implementation>
.......
</implementation>
</component>
</repository>
 Linear Solver: Application CXML

<application>
  <network>
    <instance componentName="LinearEquationSourceRowsColumnsUnsymmetric"
      componentPackage="icpc.denseLinearAlgebra.real" id="1">
      <property name="degrees of freedom" value="100"/>
    </instance>
    <instance componentName="LinearSolverRowsColumnsUnsymmetric"
      componentPackage="icpc.denseLinearAlgebra.real" id="2"/>
    <instance componentName="DisplayVector"
      componentPackage="icpc.vector.real" id="3"/>
    <dataflow sinkComponent="2" sinkPort="matrix"
      sourceComponent="1" sourcePort="matrix"/>
    <dataflow sinkComponent="2" sinkPort="vector"
      sourceComponent="1" sourcePort="vector"/>
    <dataflow sinkComponent="3" sinkPort="vector"
      sourceComponent="2" sourcePort="solution"/>
  </network>
</application>
## Local Computational Community

<table>
<thead>
<tr>
<th>System</th>
<th>Processor</th>
<th>Number</th>
<th>Language</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC (Linux)</td>
<td>AMD 900MHz</td>
<td>1</td>
<td>Java + C LAPACK</td>
<td>LU + BCG LU</td>
</tr>
<tr>
<td>Atlas</td>
<td>Alpha 667MHz</td>
<td>1,4,9</td>
<td>ScaLAPACK</td>
<td>LU + BCG</td>
</tr>
<tr>
<td>AP3000</td>
<td>UltraSparcII 300MHz</td>
<td>4,9,16</td>
<td>ScaLAPACK</td>
<td>LU + BCG</td>
</tr>
</tbody>
</table>

All processors have equal cost

## Influence of Policy on Execution Time

![Graph showing influence of policy on execution time](image)

- **Minimum Time**
- **Minimum Cost**

The graph illustrates the relationship between the number of unknowns and the execution time and cost, showing how different policies affect performance.
Influence of Policy on Execution Cost

![Graph showing minimum time and cost]

- Open XML-based Schema
- Multiple Selection Policies
- Dynamic Cost Models
- Integration with ICENI Jini-based Computational Community Grid framework
- Further use of run-time information:
  - Verification / optimisation at application level
  - Partial evaluation of implementation selections

Extensibility
Summary

• Components for E-Science
• ICENI extensible architecture
  – Layered Abstractions
  – Information for Optimisation
• Implementation selection through composite performance modelling

Acknowledgements

• Grid Middleware Group:
  – John Darlington
  – Steven Newhouse
  – Anthony Mayer
  – Nathalie Furmento
  – Stephen McGough
  – Tony Field
• Funding: EPSRC GR/N13371
• Further Information:
  – http://www-icpc.doc.ic.ac.uk/components
  – email: icpc-sw@doc.ic.ac.uk