Parallel I/O Simulator - Graphical UI Design

Samantha Dulip Withanage

February 5, 2009

Overview

1. Objectives
2. Parallel I/O Simulator - Use cases
   - Cluster
   - Use cases
3. Model
   - Component represntation (XML)
   - Component types
   - Model
4. View (GUI)
   - Interface class between Model and View
   - User interface sections
5. Achievements

Objectives of the internship

1. Use cases for the cluster - definition and documentation
2. Cluster - Model and GUI integration

Objectives of the internship

1. Use cases for the cluster - definition and documentation
2. Cluster - Model and GUI integration
3. Ease the restrictions of the available GUI

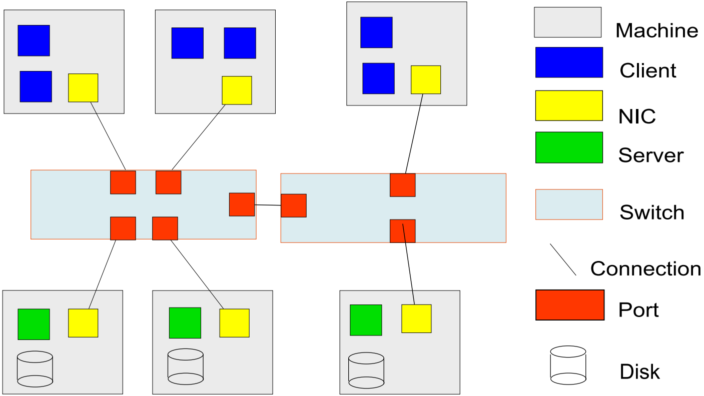
Objectives of the internship

1. Use cases for the cluster - definition and documentation
2. Cluster - Model and GUI integration
3. Ease the restrictions of the available GUI
4. Add new functionalities to the GUI

Objectives of the internship

1. Use cases for the cluster - definition and documentation
2. Cluster - Model and GUI integration
3. Ease the restrictions of the available GUI
4. Add new functionalities to the GUI

Cluster - Abstract view



Main use cases

1. Add a component
2. Remove a component

Main use cases

1. Add a component
2. Remove a component
3. Edit the attributes of component (manually / template)

Main use cases

1. Add a component
2. Remove a component
3. Edit the attributes of component (manually / template)
4. Connect a component

Main use cases

1. Add a component
2. Remove a component
3. Edit the attributes of component (manually / template)
4. Connect a component
5. Load and save a pre-defined cluster settings (global)

Main use cases

1. Add a component
2. Remove a component
3. Edit the attributes of component (manually / template)
4. Connect a component
5. Load and save a pre-defined cluster settings (global)
6. Execute programs

Main use cases

1. Add a component
2. Remove a component
3. Edit the attributes of component (manually / template)
4. Connect a component
5. Load and save a pre-defined cluster settings (global)
6. Execute programs

```
<Maschine name="CM1">
<MemorySize>1073741824</MemorySize>
<CacheSize>0</CacheSize>
<InstructionPerSecond>100</InstructionPerSecond>
<CPUs>1</CPUs>
<NIC name="MR1␣NIC1">
<Connection to="Switch␣1">
<Bandwidth>122683392</Bandwidth>
<Latency>0.00002s</Latency>
</Connection>
</NIC>
<Client name="C1" template="PVS-Client">
<Application rank="0" application="Writer1"/>
</Client>
</Maschine>
```

Component classes - definition

- BasicComponent: Parent class for all components
- NetworkComponent : NIC, Port, Connection(cable)

Component classes - definition

- BasicComponent: Parent class for all components
- NetworkComponent : NIC, Port, Connection(cable)
- Client : executes programs

Component classes - definition

- BasicComponent: Parent class for all components
- NetworkComponent : NIC, Port, Connection(cable)
- Client : executes programs
- Server : hosts I/O sub systems

Component classes - definition

- BasicComponent: Parent class for all components
- NetworkComponent : NIC, Port, Connection(cable)
- Client : executes programs
- Server : hosts I/O sub systems
- Maschine : hosts Clients, Servers, NICs

Component classes - definition

- BasicComponent: Parent class for all components
- NetworkComponent : NIC, Port, Connection(cable)
- Client : executes programs
- Server : hosts I/O sub systems
- Maschine : hosts Clients, Servers, NICs
- Switch : holds Ports

Component classes - definition

- BasicComponent: Parent class for all components
- NetworkComponent : NIC, Port, Connection(cable)
- Client : executes programs
- Server : hosts I/O sub systems
- Maschine : hosts Clients, Servers, NICs
- Switch : holds Ports

Component classes - definition

- BasicComponent: Parent class for all components
- NetworkComponent : NIC, Port, Connection(cable)
- Client : executes programs
- Server : hosts I/O sub systems
- Maschine : hosts Clients, Servers, NICs
- Switch : holds Ports

Model

Model combines all the cluster components.

- holds a component id map (CIDComponentMap)
- holds a component name map (componentNameMap)

Model

Model combines all the cluster components.

- holds a component id map (CIDComponentMap)
- holds a component name map (componentNameMap)
- holds machines list, server list, clients List and switches list

Model

Model combines all the cluster components.

- holds a component id map (CIDComponentMap)
- holds a component name map (componentNameMap)
- holds machines list, server list, clients List and switches list
- holds the template manager

Model

Model combines all the cluster components.

- holds a component id map (CIDComponentMap)
- holds a component name map (componentNameMap)
- holds machines list, server list, clients List and switches list
- holds the template manager

Class Component2D

Component2D is the graphical extraction of the BasicComponent

1. is an abstract representation of a component
2. maps the BasicComponent to the 2D version

Class Component2D

Component2D is the graphical extraction of the BasicComponent

1. is an abstract representation of a component
2. maps the BasicComponent to the 2D version
3. extends the basic JPanel (uniform for each component)

Class Component2D

Component2D is the graphical extraction of the BasicComponent

1. is an abstract representation of a component
2. maps the BasicComponent to the 2D version
3. extends the basic JPanel (uniform for each component)
4. defines the basic painting behavior

Class Component2D

Component2D is the graphical extraction of the BasicComponent

1. is an abstract representation of a component
2. maps the BasicComponent to the 2D version
3. extends the basic JPanel (uniform for each component)
4. defines the basic painting behavior
5. is extended by each cluster component

Class Component2D

Component2D is the graphical extraction of the BasicComponent

1. is an abstract representation of a component
2. maps the BasicComponent to the 2D version
3. extends the basic JPanel (uniform for each component)
4. defines the basic painting behavior
5. is extended by each cluster component

Class Component2D

Component2D is the graphical extraction of the BasicComponent

1. is an abstract representation of a component
2. maps the BasicComponent to the 2D version
3. extends the basic JPanel (uniform for each component)
4. defines the basic painting behavior
5. is extended by each cluster component

User interface sections

User interface consists of three main sections.

1. Menu Bar
2. Drawing Area

User interface sections

User interface consists of three main sections.

1. Menu Bar
2. Drawing Area
3. Left Panel

User interface sections

User interface consists of three main sections.

1. Menu Bar
2. Drawing Area
3. Left Panel

Menu bar

All the cluster related tasks are accessible in the menu bar.

1. Project (New, Open, Save, Close)
2. Components

Menu bar

All the cluster related tasks are accessible in the menu bar.

1. Project (New, Open, Save, Close)
2. Components
3. Tools (Template manager.)

Menu bar

All the cluster related tasks are accessible in the menu bar.

1. Project (New, Open, Save, Close)
2. Components
3. Tools (Template manager.)
4. Simulation

Menu bar

All the cluster related tasks are accessible in the menu bar.

1. Project (New, Open, Save, Close)
2. Components
3. Tools (Template manager.)
4. Simulation
5. Help

Menu bar

All the cluster related tasks are accessible in the menu bar.

1. Project (New, Open, Save, Close)
2. Components
3. Tools (Template manager.)
4. Simulation
5. Help

Drawing area

Components are drawn in the canvas called, drawing area.

1. Three virtual sections
2. Top section: maschines (clients)

Drawing area

Components are drawn in the canvas called, drawing area.

1. Three virtual sections
2. Top section: maschines (clients)
3. Middle section: switches

Drawing area

Components are drawn in the canvas called, drawing area.

1. Three virtual sections
2. Top section: maschines (clients)
3. Middle section: switches
4. Bottom section : maschines (servers)

Drawing area

Components are drawn in the canvas called, drawing area.

1. Three virtual sections
2. Top section: maschines (clients)
3. Middle section: switches
4. Bottom section : maschines (servers)

Drawing area

Components are drawn in the canvas called, drawing area.

1. Three virtual sections
2. Top section: maschines (clients)
3. Middle section: switches
4. Bottom section : maschines (servers)

Left Panel

Component related tasks and global settings are listed in the left panel.

1. dynamical activation
2. add a new component
3. delete a selected component

Left Panel

Component related tasks and global settings are listed in the left panel.

1. dynamical activation
2. add a new component
3. delete a selected component
4. change the attributes of a component

Left Panel

Component related tasks and global settings are listed in the left panel.

1. dynamical activation
2. add a new component
3. delete a selected component
4. change the attributes of a component
5. select the templates

Left Panel

Component related tasks and global settings are listed in the left panel.

1. dynamical activation
2. add a new component
3. delete a selected component
4. change the attributes of a component
5. select the templates
6. change global settings

Left Panel

Component related tasks and global settings are listed in the left panel.

1. dynamical activation
2. add a new component
3. delete a selected component
4. change the attributes of a component
5. select the templates
6. change global settings

Left Panel

Component related tasks and global settings are listed in the left panel.

1. dynamical activation
2. add a new component
3. delete a selected component
4. change the attributes of a component
5. select the templates
6. change global settings

Achievements

Color code : Completed , minor issues, Open

1. Usability definition and documentation
2. Redesigning of the graphical user interface

Achievements

Color code : Completed , minor issues, Open

1. Usability definition and documentation
2. Redesigning of the graphical user interface
3. Integration of the model/gui

Achievements

Color code : Completed , minor issues, Open

1. Usability definition and documentation
2. Redesigning of the graphical user interface
3. Integration of the model/gui
4. Add/Edit/delete component (connection)

Achievements

Color code : Completed , minor issues, Open

1. Usability definition and documentation
2. Redesigning of the graphical user interface
3. Integration of the model/gui
4. Add/Edit/delete component (connection)
5. Load and save pre-defined cluster settings

Achievements

Color code : Completed , minor issues, Open

1. Usability definition and documentation
2. Redesigning of the graphical user interface
3. Integration of the model/gui
4. Add/Edit/delete component (connection)
5. Load and save pre-defined cluster settings
6. overlapping of the connections

Achievements

Color code : Completed , minor issues, Open

1. Usability definition and documentation
2. Redesigning of the graphical user interface
3. Integration of the model/gui
4. Add/Edit/delete component (connection)
5. Load and save pre-defined cluster settings
6. overlapping of the connections

Demo

Thank you!