



## Outline of the presentation

- Overview
- Need for an IDE
- Introduction to Grid IDE
- Components
- Features
  - Project Development and Management
  - Resource Management
  - Job Management
  - Debugger
  - Profiler
  - Help
- Conclusion

## Outline of the presentation

- **Overview**
- Need for an IDE
- Introduction to Grid IDE
- Components
- Features
  - Project Development and Management
  - Resource Management
  - Job Management
  - Debugger
  - Profiler
  - Help
- Conclusion

## Overview

- Grid applications
  - Bioinformatics applications,
  - Disaster management applications,
  - Neuroscience applications,
  - Atmospheric science and Climate Modeling, ...
- Users
  - Researchers,
  - Scientists,
  - Engineers, ...

# Knowledge of Grid computing among users

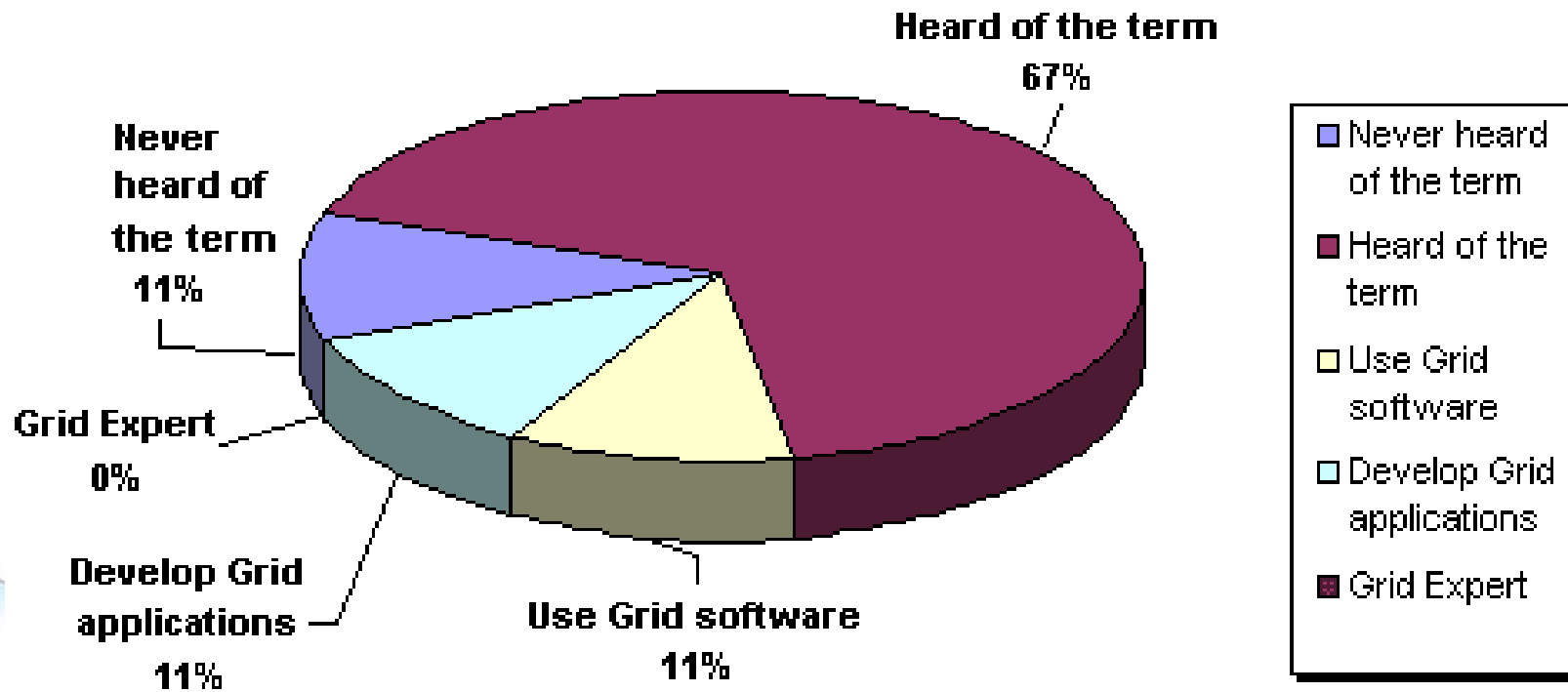


Chart illustrating knowledge of Grid computing in the research community

## Outline of the presentation

- Overview
- **Need for an IDE**
- Introduction to Grid IDE
- Components
- Features
  - Project Development and Management
  - Resource Management
  - Job Management
  - Debugger
  - Profiler
  - Help
- Conclusion

## Need for an IDE – Existing Problems

- Command line tools pose difficulties for the users.
- Need to memorize command line options.
- Shuttling between an editor and the compiler during compilation and debugging of an application.
- Need for source-level Debugger.
- Need for a Profiler.

## Need for an IDE - Solutions

- Researchers should become grid experts.
  - For which they may not be interested.
- Outsource grid applications for execution.
  - Which may not be possible due to security reasons.
- Use a single interface with all the tools required for developing an application.
  - Ex: **Grid IDE**

## Outline of the presentation

- Overview
- Need for an IDE
- **Introduction to Grid IDE**
- Components
- Features
  - Project Development and Management
  - Resource Management
  - Job Management
  - Debugger
  - Profiler
  - Help
- Conclusion

## Introduction to Grid IDE

- Single convenient environment with various application development tools.
- Enable users to carry out entire program development life cycle for the Grid.
- Improves the efficiency and productivity of the developer.
- Narrow the gap between Grid users and Grid tools providers.

## Features of Grid IDE

- User friendly
- Developing and enabling applications to exploit the Grid.
- Supports multiple programming interfaces.
- Grid applications will be Globus based,
  - No need to know Globus
  - No need to write Globus code
  - No need to rewrite legacy code

## Features of Grid IDE

- Best utilization of available resources as per user's choice and within their budget.
- Compiling across different platforms.
- Helps to Develop, Build, Debug, Profile grid applications.

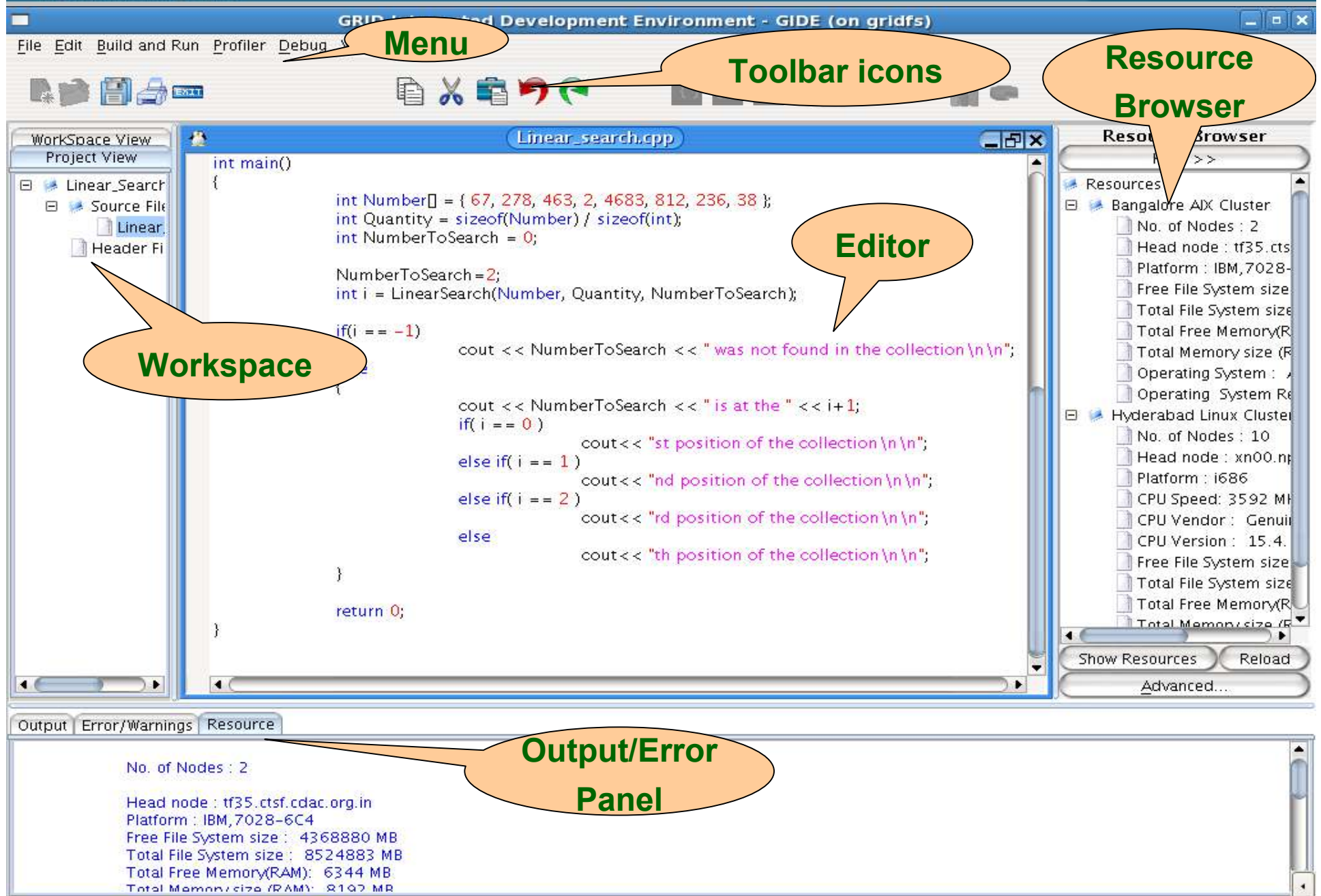
## Outline of the presentation

- Overview
- Need for an IDE
- Introduction to Grid IDE
- **Components**
- **Features**
  - Project Development and Management
  - Resource Management
  - Job Management
  - Debugger
  - Profiler
  - Help
- Conclusion

## Components of Grid IDE

- ✓ Project Development and Management
- ✓ Resource Management
- ✓ Job Management
- ✓ Inbuilt source-level Debugger
- ✓ Profiler
- Accounting
- ✓ Help

# Snapshot of Grid IDE



The screenshot displays the Grid IDE interface with several components highlighted by callouts:

- Menu:** Located at the top left, containing options like File, Edit, Build and Run, Profiler, and Debug.
- Toolbar icons:** A row of icons for file operations (save, print, copy, paste) and development actions (run, refresh, undo, redo).
- Resource Browser:** A panel on the right showing a tree view of resources, including 'Bangalore AIX Cluster' and 'Hyderabad Linux Cluster', with details like 'No. of Nodes' and 'Head node'.
- Editor:** The central workspace showing a C++ program named 'Linear\_search.cpp' with code for a linear search algorithm.
- Workspace:** A panel on the left showing a project view with folders for 'Linear\_Search', 'Source File', and 'Header File'.
- Output/Error Panel:** A panel at the bottom showing the output of the program, including system information like 'No. of Nodes : 2' and 'Head node : tf35.ctsf.cdac.org.in'.

## Outline of the presentation

- Overview
- Need for an IDE
- Introduction to Grid IDE
- Components
- **Features**
  - Project Development and Management
  - Resource Management
  - Job Management
  - Debugger
  - Profiler
  - Help
- Conclusion

## Features of GUI

- User-friendly – easy to use and handle
- Supports multiple editor windows.
- Options are provided through Menus, Shortcut Icons, Accelerator and Hot keys.
- **Display panel** to display Output/ Error/ Grid resources.

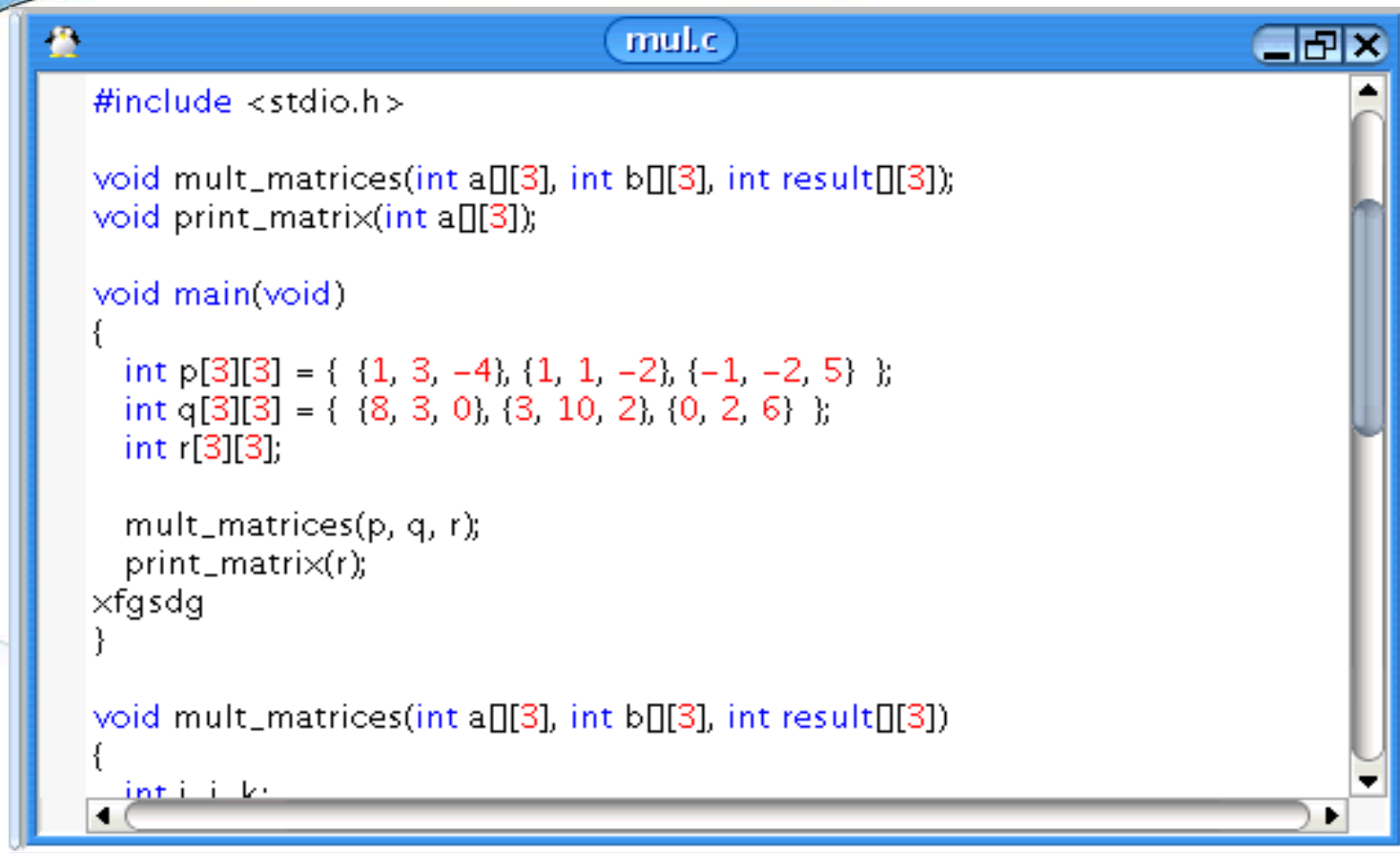
## Outline of the presentation

- Overview
- Need for an IDE
- Introduction to Grid IDE
- Components
- Features
  - **Project Development and Management**
  - Resource Management
  - Job Management
  - Debugger
  - Profiler
  - Help
- Conclusion

## Features of Project Development & Management

- Basic File and Project management features - Adding / Remove any file from the project.
- Concept of Workspace.
- User-defined Library Management Module.
- Inbuilt Editor
  - Supports C, C++, Fortran
  - Keyword coloring
  - Unlimited Undo-Redo, Cut, Copy and Paste
- Automatic Makefile and RSL file generator.

# Features of Project Development & Management



```
#include <stdio.h>

void mult_matrices(int a[][3], int b[][3], int result[][3]);
void print_matrix(int a[][3]);

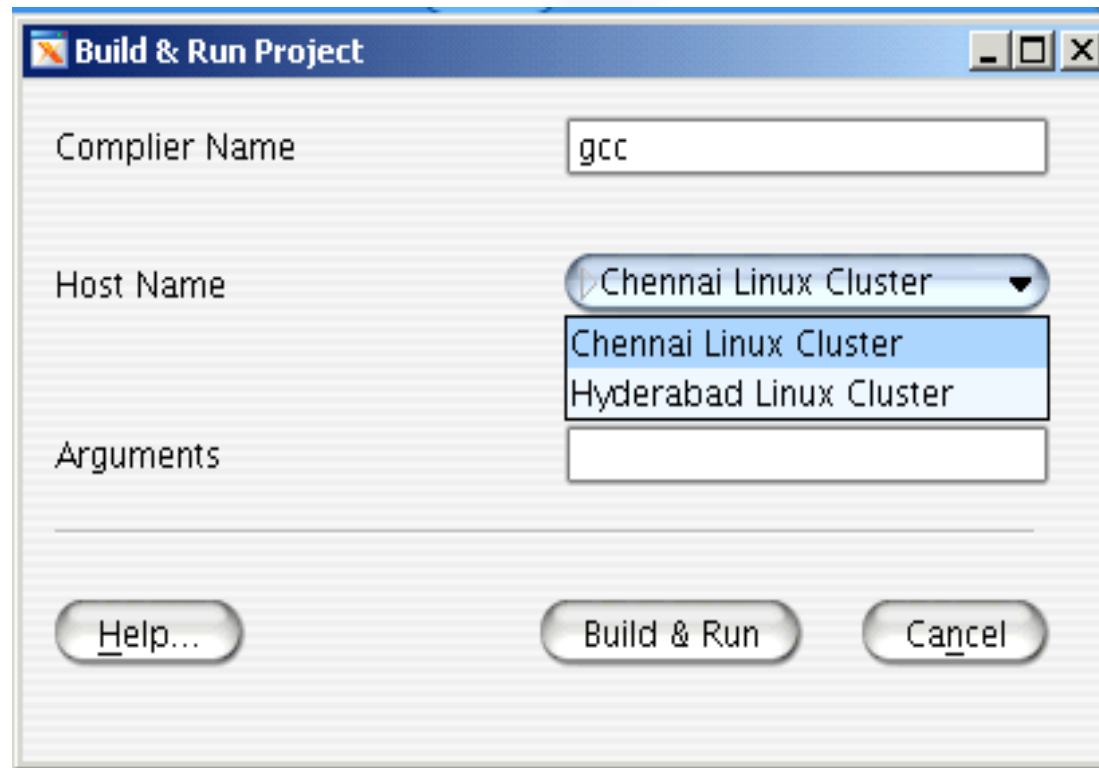
void main(void)
{
    int p[3][3] = { {1, 3, -4}, {1, 1, -2}, {-1, -2, 5} };
    int q[3][3] = { {8, 3, 0}, {3, 10, 2}, {0, 2, 6} };
    int r[3][3];

    mult_matrices(p, q, r);
    print_matrix(r);
    xfgsdg
}

void mult_matrices(int a[][3], int b[][3], int result[][3])
{
    int i, j, k;
```

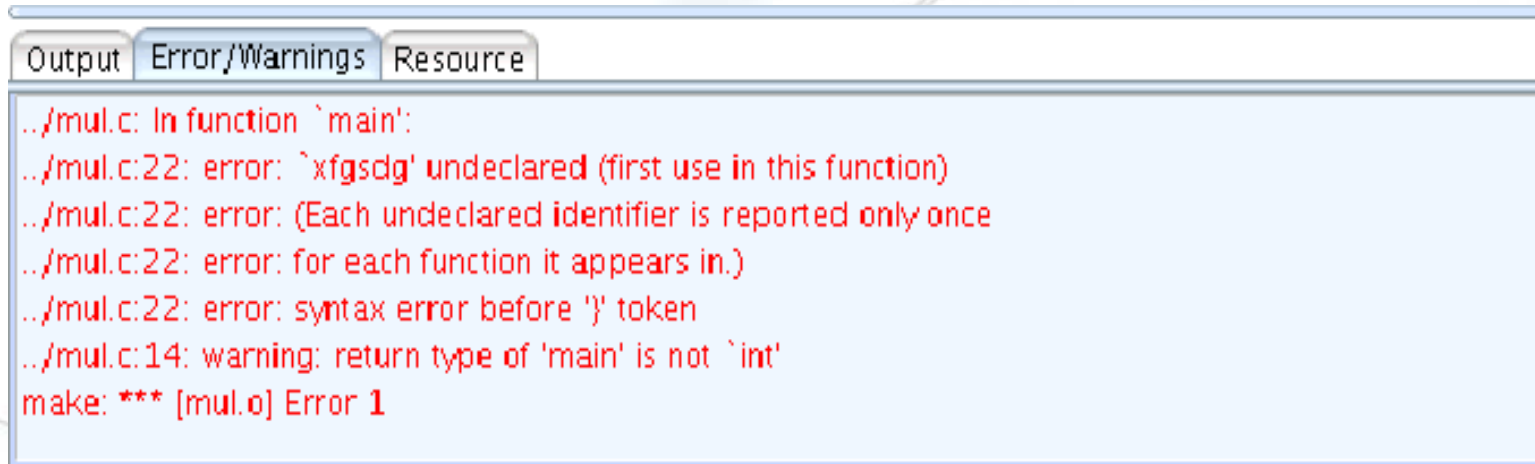
# Features of Project Development & Management

- Build/ Execute/ Compile on selected resources.



# Features of Project Development & Management

- Display compilation/ execution errors from each node to the user.



```
Output Error/Warnings Resource
../mul.c: In function `main':
../mul.c:22: error: `xfgsdg' undeclared (first use in this function)
../mul.c:22: error: (Each undeclared identifier is reported only once
../mul.c:22: error: for each function it appears in.)
../mul.c:22: error: syntax error before `}' token
../mul.c:14: warning: return type of `main' is not `int'
make: *** [mul.o] Error 1
```

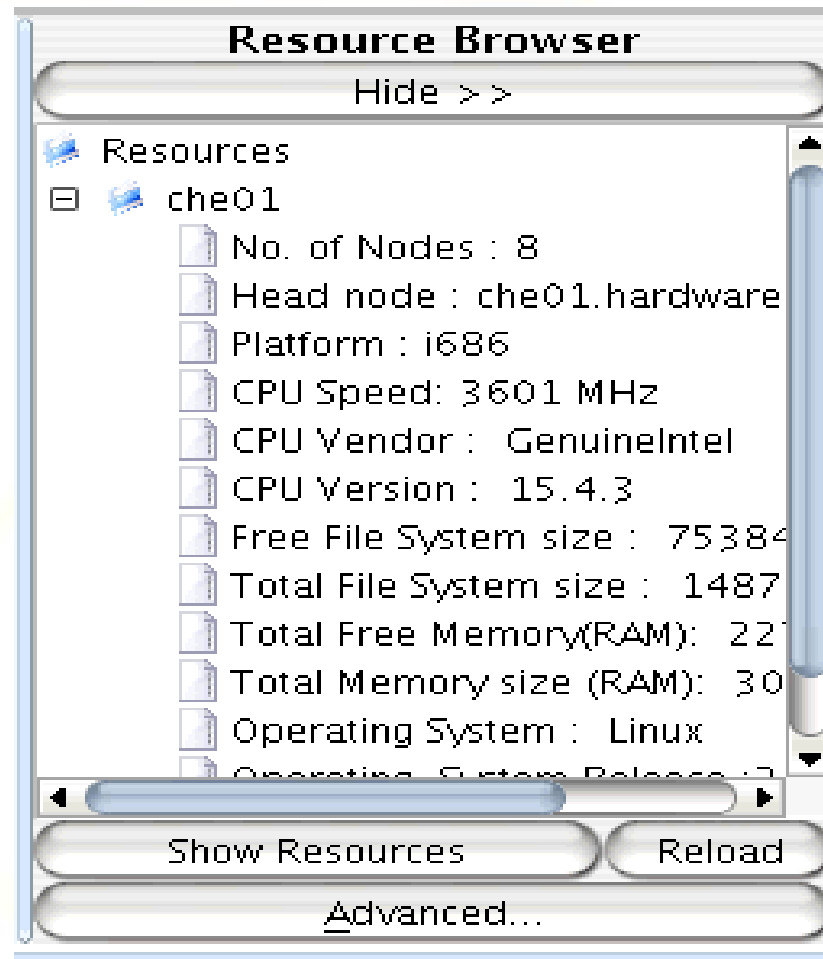
## Outline of the presentation

- Overview
- Need for an IDE
- Introduction to Grid IDE
- Components
- Features
  - Project Development and Management
  - **Resource Management**
  - Job Management
  - Debugger
  - Profiler
  - Help
- Conclusion

## Features of Resource Management

- Displays cluster components details:
  - Number of nodes, Head Node
  - CPU – speed, vendor, version
  - Total and Free Memory available
  - Operating system ...
- Dynamic updating of information of resources.
- Advanced Search option to facilitate selection of resources based on
  - Operating Systems
  - Platforms ...

# Features of Resource Management



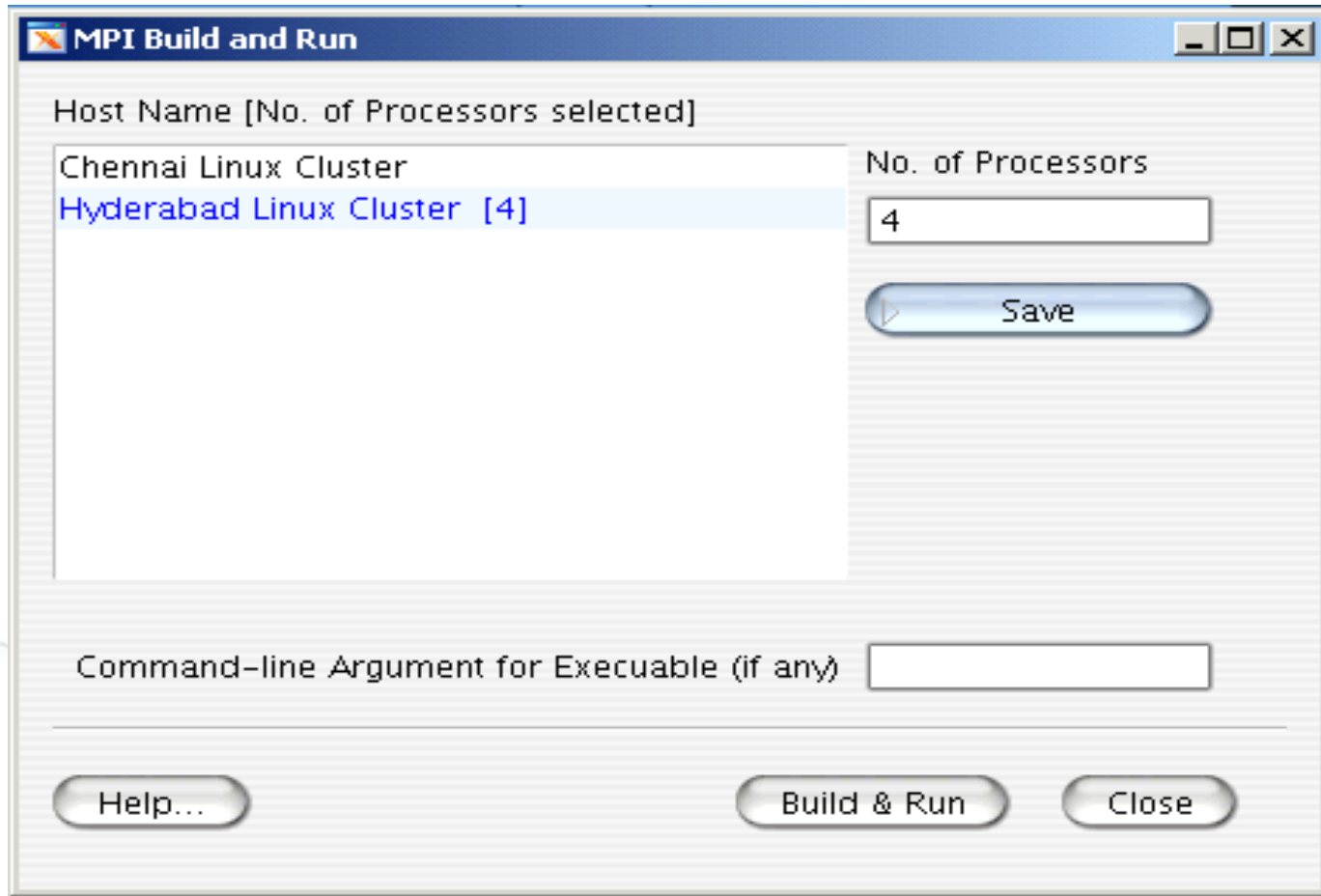
## Outline of the presentation

- Overview
- Need for an IDE
- Introduction to Grid IDE
- Components
- Features
  - Project Development and Management
  - Resource Management
  - **Job Management**
  - Debugger
  - Profiler
  - Help
- Conclusion

## Features of Job Management

- Building and Executing (Serial and Parallel) applications on local machine and on Grid.
- Allows compilation of single file of a project.
- Allows user to select multiple CPUs on multiple nodes for parallel applications.
- Job submission made easy with mouse interface.

# Features of Job Management



## Outline of the presentation

- Overview
- Need for an IDE
- Introduction to Grid IDE
- Components
- Features
  - Project Development and Management
  - Resource Management
  - Job Management
  - **Debugger**
  - Profiler
  - Help
- Conclusion

## Features of Debugger

- Supports debugging at source level.
- Debugging both serial and parallel applications.
- Supports standard debugging features:
  - Single stepping - Step in, Step, out, Step over
  - Break at – Function call or Line number
  - Run to cursor
  - Execute till completion
  - Insert/ Remove breakpoints
  - Display value or type of variables by pointing the mouse on the variable...

# Features of Debugger

```
mul.c
** multiply two 3X3 matrices.
**
** Justin, Feb 20, '97
*/
#include <stdio.h>

void mult_matrices(int a[][3], int b[][3], int result[][3]);
void print_matrix(int a[][3]);

int main(void)
{
    int p[3][3] = { {1, 3, -4}, {1, 1, -2}, {-1, -2, 5} };
    int q[3][3] = { {8, 3, 0}, {3, 10, 2}, {0, 2, 6} };
    int r[3][3];
}

Output Error/Warnings Resource
Breakpoint 1, main () at ../mul.c:15
□□/home/dag07/GIDE/Matrix_MUL/mul.c:15:207: beg:0x8048389
(GIDE)
□□/home/dag07/GIDE/Matrix_MUL/mul.c:16:267: beg:0x804839b
(GIDE)
```

# Features of Debugger

Workshop on Developing Applications on Grid - GARUDA

**mul.c**

```

**
** Justin, Feb 20, '97
**/
#include <stdio.h>

void mult_matrices(int a[][3], int b[][3], int result[3][3]);
void print_matrix(int a[][3]);

int main(void)
{
    int p[3][3] = { {1, 3, -4}, {1, 1, -2}, {-1, -2, 5} };
    int q[3][3] = { {8, 3, 0}, {3, 10, 2}, {0, 2, 6} };
    int r[3][3];

    mult_matrices(p, q, r);
}

```

**BreakPoint Information**

Num	Type	Disp	Enb	Address	What
1	breakpoint	keep	y	0x08048389	in main at ../mu
2	breakpoint	keep	y	0x080483ad	in main at ../mu

**Output**

```

(GIDE)
Num Type      Disp Enb Address  What
1 breakpoint  keep y  0x08048389 in main at ../mul.c:15
              breakpoint already hit 1 time
2 breakpoint  keep y  0x080483ad in main at ../mul.c:19
(GIDE)

```

National Grid Computing Initiative - GARUDA

## Features of Debugger

- Local debugging support for serial applications.
- Debugging of each process on each node for parallel applications.

## Outline of the presentation

- Overview
- Need for an IDE
- Introduction to Grid IDE
- Components
- Features
  - Project Development and Management
  - Resource Management
  - Job Management
  - Debugger
  - **Profiler**
  - Help
- Conclusion

## Features of Profiler

- Depicts time consumed by different functions or subroutines of the application graphically.
- Information about the total time spent in the execution of application.
- Useful in identifying the parts of the program which needs modification for achieving better performance.

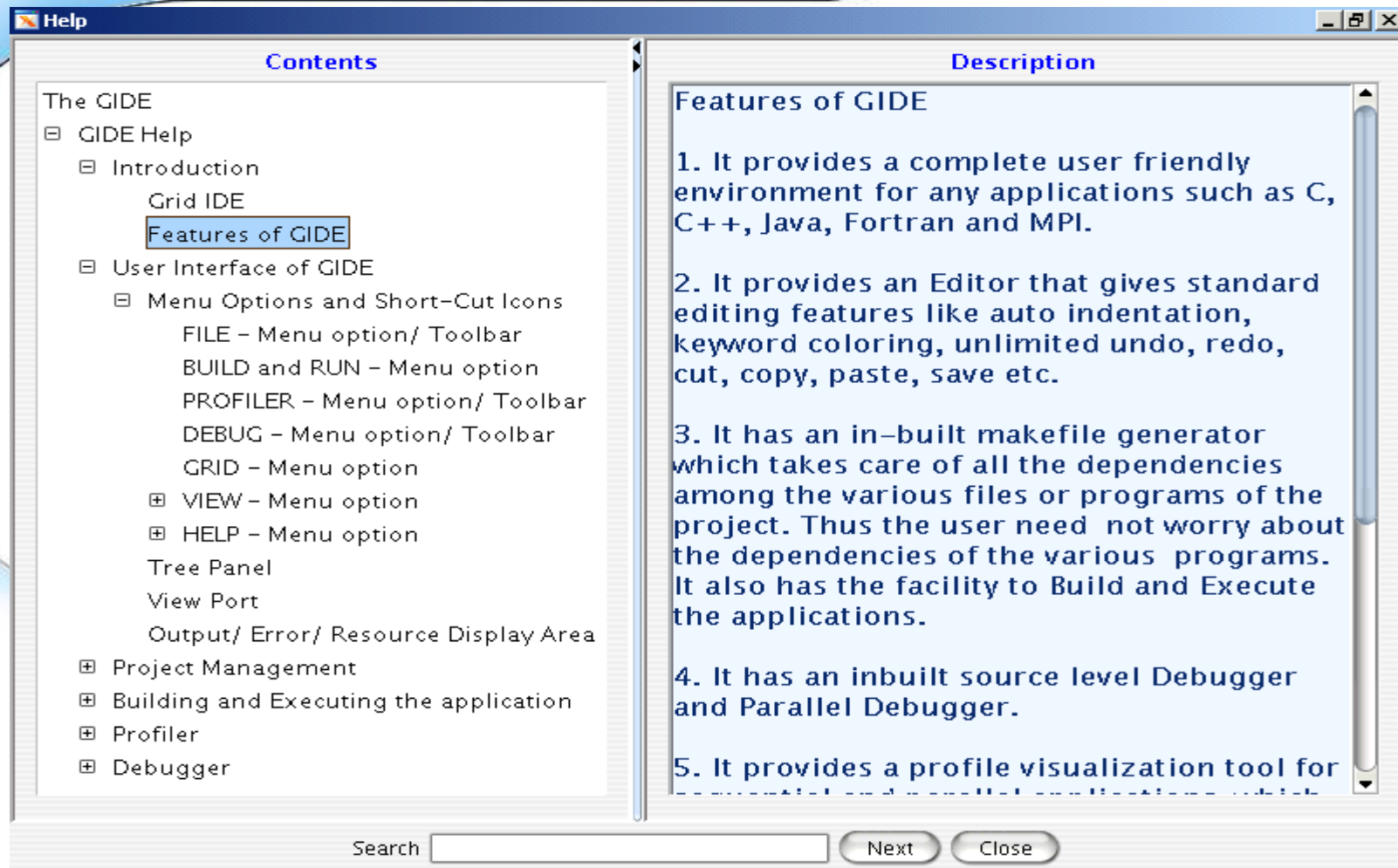
## Outline of the presentation

- Overview
- Need for an IDE
- Introduction to Grid IDE
- Components
- Features
  - Project Development and Management
  - Resource Management
  - Job Management
  - Debugger
  - Profiler
  - **Help**
- Conclusion

## Features of Online Help

- Help button provides information about various screen of Grid IDE.
- Provides Context sensitive help.
- Supports indexed help for faster access.

# Features of Online Help



The screenshot shows a window titled 'Help' with two panes: 'Contents' and 'Description'. The 'Contents' pane shows a tree view where 'Features of GIDE' is selected. The 'Description' pane displays the following text:

**Features of GIDE**

1. It provides a complete user friendly environment for any applications such as C, C++, Java, Fortran and MPI.
2. It provides an Editor that gives standard editing features like auto indentation, keyword coloring, unlimited undo, redo, cut, copy, paste, save etc.
3. It has an in-built makefile generator which takes care of all the dependencies among the various files or programs of the project. Thus the user need not worry about the dependencies of the various programs. It also has the facility to Build and Execute the applications.
4. It has an inbuilt source level Debugger and Parallel Debugger.
5. It provides a profile visualization tool for

At the bottom of the window, there is a search bar and 'Next' and 'Close' buttons.

## Outline of the presentation

- Overview
- Need for an IDE
- Introduction to Grid IDE
- Components
- Features
  - Project Development and Management
  - Resource Management
  - Job Management
  - Debugger
  - Profiler
  - Help
- Conclusion

## Conclusion

### Grid IDE ...

- Provides standard environment across the GARUDA Grid.
- Enable users to carry out entire program development life cycle on the Grid.
- Single interface supporting –
  - Basic Program Development Tools – Editor, Compiler;
  - Program Analysis Tools – Debugger, Profiler.
- Increases efficiency and productivity of the developer.

Any questions?

Thank you!

Advanced Computing For Human Advancement

