Alchemi.NET Framework in Grid Computing

Vijay Dhir
Lecturer, Dept. of Computer Science & Engineering, S.B.B.S.I.E.T, Padiana
Distt: Jallandhar. E-Mail: vijaydhir@ymail.com

[Acknowledgment: The Research work is done under the guidance of my guide Dr. Rattan K. Datta]

Abstract:
Computer Technology has Revolutionized Science. Scientists have developed accurate mathematical model of Physical universe and computer programmed with these models can approximate reality at many levels of scale: an atomic nucleus, weather forecasting, the earth’s biosphere or the entire Universe. We can predict the future validate or disapprove theories using these programs. If the computer power is greater the greater will be the accuracy in approximation i.e close will be the approximation to the reality. The speed of the computer should be fast enough to do calculations in the small amount of time. The way to speed up the computation is to “parallize” it i.e. divide the work into pieces that can be worked on by separate processors at the same time. Thus we can solve the problems that are Non-Polynomial form in polynomial time. If the Scientist has an idea for a computation, but finds that it will take a million years of computer time, normal reaction is to toss the idea in a Waste bucket. One of the approach is to use multimillion dollar Supercomputer e.g. Jaguar by Cray Research Inc and Installed at Oak Ridge Nation and Eka (fourth fast Super Computer) by Tata or Computational Grid that are geographically distributed resources e.g. SETI@home (Used to detect radio waves emitted by intelligent civilizations outside earth) has 4.6 million participants computers. So instead of investing so much of money on Super Computer, we can built Computational grid having the power same that of the desktop computer. Computational Grid is used in number of applications like Weather Forecasting, Climate Change Prediction, Detection of Radio Signals emitted by intelligent Civilizations outside earth etc. In this paper the Alchemi.NET framework is used for calculating the 100 digit places value after decimal of Pi (π) value.

1. Introduction:
Technology have developed a lot. According to Moore’s Law [1], the Complexity of the chip doubles every 18 months. Despite the exponential increase in the processing power the scientists still need super fast computers to solve some of the complex problems. The Foster-Kesselman duo organized in 1997, at Argonne National Laboratory, a workshop entitled “Building a Computational Grid”. At this moment the term “Grid” was born. The workshop was followed in 1998 by the publication of the book “The Grid: Blueprint for a New Computing Infrastructure” by Foster and Kesselman [3] themselves. For these reasons they are considered the fathers of the Grid.

Grid computing very promising many independent computer are grouped together to solve a particular complex problem. According to Foster-Kesselman [4] “ Grid Computing is a special type of parallel Computing which relies on complete computers (with onboard CPU, Power supply, network interface etc) connected to the internet by conventional network interface, such as Ethernet”. “Grid Computing” is a special type of parallel computing which relies on complete computers (having own CPU, Storage, NIC, Power supply etc.) connected to the internet by the conventional network interface, such as Ethernet. Grid Computer concept comes up during 1990’s as a part of high-performance computing: - In Grid computing the individual nodes within a Grid are not specialized, dedicated components. The individual system is based on standard machines and operating system. It is the application software on the top of the OS, that supports the Grid functionality.

So we can think Grid computer concept as a virtual super computer at a very low cost. Desktop PCs in corporate and home environments are under utilized-typically one tenth of the processing power is used. So, Unused CPU cycles are of PC are used over the Internet to generate the power of supercomputer. So we can say by using this concept we can build " Poor man’s Super Computer".

We do not need any special infrastructure like homogeneous computers. Large Memory. Fast Processor etc. Grid works with heterogeneous as well as homogeneous computer we can connect as may computers as we can along the chain of computer which will given that free cycles of processing to solve complex problems. Various software tool bits are available to build such a Grid environment Among these, the Globus Toolkit has been widely used in research all over the world for longer than a decade Alchemi is the latest addition to Grid Software apart from above some other toolkits are Globus toolkit, Condor, Netsolve etc. There are basically two types of Grid based on their OS like Linux based desktop Grid computing and windows based desktop Grid computing. Majority of desktop computers run variants of windows operating systems. So in this paper windows based desktop, grid computer is considered for the above purpose, windows-
based desktop grid computing framework called Alchemi implemented on the Microsoft .NET platform. Computer technology has revolutionized science. Scientists have developed accurate mathematical models of physical universe and computers programmed with these models can approximate reality at many levels of scale an atomic nucleus, weather forecasting, the earth biosphere or the entire universe. We can predict the future, validate or disapprove theories. Using these programs.

If the computer power is greater the greater will be the accuracy in approximation i.e close will be the approximation to reality. Also the speed of the computer should be fast enough to do millions of calculation in a small amount of time. The way to speed up the computation is to “parallelize” it to divide the work into pieces that can be worked on by separate processors at the same time. Thus we can solve the compiler problems that are non-polynomial form in polynomial time. Technology have developed a lot. According to Moore’s law [1], the speed of chip doubles in every 18 months it is because of Moore’s law. PC’s become more fast and there are millions of fast computer connected through Internet. The idea of using these computers as grid. SETI@home [2] has been very successful in this regard. It has 4.6 million participants, out of which 600,000 remain active. The goal of this project is detecting radio signal emitted by intelligent civilization outside earth.

2. Literature review of desktop Grid:
Alchemi is the .NET framework that provides runtime machinery and programming environment requires constructing desktop Grid. It supports object oriented programming in addition to file based job model cross platform is provided by web server interface. Entropia [9] uses a window desktop grid system by aggregating the raw desktop resources into a single logical resource. There is a centralized computer, which administrates various desktop clients. But it does not provide web server interface for cross platform. SETI@home [2] is only application specific. The Extraterrestrial Intelligence (SETI) is a university of California project. It developed desktop grid system that has hundred and thousand of PC’s across Internet to process massive amount of astronomy data capturing through Arecibo telescope based at Puerto Rico every day. The module is designed as application specific software only. Condor [10] system is developed by university of Wisconsin at Madison. Unique mechanism enable condor to effectively harness wasted CPU Power from idle desktop workstations. Uses submit their job to condor, condor places them into a queue, chooses when and where to run the jobs based upon a policy monitors is the progress and informs upon completion. It can handle both windows and Unix class resources in its resource pools. This does not have thread-programming model and does not support cross platform web services interface.

<table>
<thead>
<tr>
<th></th>
<th>Alchemi</th>
<th>Cond or</th>
<th>SETI@home</th>
<th>Entropia</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture</td>
<td>Hierarchical</td>
<td>Hierarchical</td>
<td>Centralized</td>
<td>Centralized</td>
</tr>
<tr>
<td>Web Services Interface for cross-platform Integration</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Implementa tion Technologies</td>
<td>C# Web services &amp; .NET Framework</td>
<td>C</td>
<td>C++, Win 32</td>
<td>C++, Win 32</td>
</tr>
<tr>
<td>Thread programmin g Model</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Level of Integration of application, Prog. &amp; runtime Environment</td>
<td>Low (General Purpose)</td>
<td>Low (General Purpose)</td>
<td>High (Single Application Environment)</td>
<td>Low (General Purpose)</td>
</tr>
</tbody>
</table>

submitting them to the manager for execution by Grid.
3.1 Distributed components:
Four Types of modes take part in desktop Grid construction and application execution. An Alchemi desktop Grid is constructed by deploying a manager node and deploying one or more Executors nodes configured to connect to the Manager one or more users can execute their application on the cluster by connecting to the manager. An optional component, the cross platform Manager provides a web services interface to custom Grid middleware.

3.1.1 Manager:
Executors gets register with the manager users sends threads to the manager which are placed in a pool and scheduled to be executed on the various available Executors. Manager Monitor the status of Executors Job submitted and completed, the priority for the each thread can be specified when it is created or submitted. The executors return the threads after completion to the Manager which are subsequently collected by respective user. The Manager employ role based security model for authentication and authorization of secure activities.

3.1.2 Executor:
The Executor Accepts the thread from Manager and executes them. An Executor can be dedicated and nondedicated. Dedicated means the resource is centrally managed by the manager and non-dedicated means that resource is managed on a volunteer basis via screen saver or explicitly by a user. Dedicated executor is more suitable where the Manager and Executor are on the same local Area Network while non Dedicated execution is more appropriate when the manager and executor are to be connected over the internet. All Grid threads run in the background with the lowest priority. Thus any uses programs are unaffected since they have higher priority access to the CPU over Grid threads.

3.1.3 User:
Grid Applications are executed on the uses mode. The API Abstracts the implementation of the Grid from the uses and is responsible for performing a variety of services on the user's behalf such as submitting an application and its constituent threads for execution, notifying the user of finished threads and providing results and notifying the user of failed threads along with error details etc.

3.1.4 Cross platform Manager:
Alchemi Provides a cross platform web service interface[11][12] to the manager. This can be sued by software written on the other platforms to extend Alchemi Grids. Jobs submitted to the cross platform Manager are translated into a firm that is acceptable by the manager which are the scheduled and executed as normal fashion.

4. Architecture of Alchemi.NET:
A grid application that has to be executed on grid consists of the number of grid threads. Grid thread is the smallest unit of parallel execution. Alchemi follows Master- worker parallel programming paradigm[8] in which central component dispatches independent units of parallel execution to workers and manages them. The diagram is as shown in figure 4.3.
5. Life Cycle of Grid Application
To develop and execute a grid application the developer creates a custom grid thread class that derives from the abstract GThread class. An instance of the GApplication object is created and any dependencies required by the application are added to its DependencyCollection. Instances of the GThread derived class are then added to the GApplication’s ThreadCollection. The GApplication serializes and sends relevant data to the Manager, where it is persisted to disk and threads scheduled. Application and thread state is maintained in a SQL Server / MSDE database. Figure shows the interaction between the executor and manager nodes. Figure 5.1 shows interaction between executor and Manager nodes.

6. Installation, Configuration and Operation
This section describes the installation, configuration and operation of the various parts of the framework for setting up Alchemi grids.
• Microsoft .NET [5] Framework 1.1
6.2. Manager
The Manager should be installed on a stable and reasonably capable machine. The Manager requires:
• SQL Server 2000 or MSDE 2000
6.2.1 Installation
The Alchemi Manager can be installed[16] in two modes
o As a normal Windows desktop application
o As a windows service. (supported only on Windows NT/2000/XP/2003)
6.3 Configuration & Operation

- The Manager can be run from the desktop or Start -> Programs -> Alchemi -> Manager -> Alchemi Manager. The database configuration settings used during installation automatically appear when the Manager is first started.
- Click the "Start" button to start the Manager.
- When closed, the Manager is minimised to the system tray.

Figure 6.2. Manager minimised to system tray.

Manager Logging

The manager logs its output and errors to a log file called "alchemi-manager.log". This can be used to debug the manager / report errors / verify the manager operation. The log file is placed in the “dat” directory under the installation directory.

6.4. Role-Based Security

Every program connecting to the Manager must supply a valid username and password. Three default accounts are created during installation: executor (password: executor), user (password: user) and admin (password: admin) belonging to the 'Executors', 'Users' and 'Administrators' groups respectively. Users are administered via the 'Users' tab of the Alchemi Console (located in the Alchemi SDK). Only Administrators have permissions to manage users; you must therefore initially log in with the default admin account. The Console lets you add users, modify their group membership and change passwords. The Users group (grp_id = 3) is meant for users executing grid applications. The Executors group (grp_id = 2) is meant for Alchemi Executors. By default, Executors attempting to connect to the Manager will use the executor account. If you do not wish Executors to connect anonymously, you can change the password for this account.
You should change the default admin password for production use.

6.5 Cross Platform Manager
The Cross Platform Manager (XPManager) requires:
- Internet Information Services (IIS)
- ASP.NET
Installation
- Install the XPManager web service via the Cross Platform Manager installer.
Configuration
- If the XPManager is installed on a different machine that the Manager, or if the default port of the Manager is changed, the web service's configuration must be modified. The XPManager is configured via the ASP.NET Web.config file located in the installation directory (wwwroot\Alchemi\CrossPlatformManager by default):

6.6 Executor
Installation
- The Alchemi Executor can be installed in two modes
  o As a normal Windows desktop application
  o As a windows service. (supported only on Windows NT/2000/XP/2003)
- To install the executor as a windows application, use the Executor Setup installer. For servicemode installation use the Executor Service Setup. The configuration steps are the same for both modes. In case of the service-mode, the “Alchemi Executor Service” installed and configured to run automatically on Windows start-up. After installation, the standard Windows service control manager can be used to control the service. Alternatively the Alchemi ExecutorServiceController program can be used. The Executor service controller is a graphical interface, which looks very similar to the normal Executor application.
  - Install the Executor via the Executor installer and follow the on-screen instructions.
Configuration & Operation
The Executor can be run from the desktop or Start -> Programs -> Alchemi -> Executor -> Alchemi Executor.
The Executor is configured from the application itself. You need to configure 2 aspects of the Executor:
  - The host and port of the Manager to connect to.
  - Dedicated / non-dedicated execution. A non-dedicated Executor executes grid threads on a voluntary basis (it requests threads to execute from the Manager), while a dedicated Executor is always executing grid threads (it is directly provided grid threads to execute by the Manager). A non-dedicated Executor works behind firewalls.
  - Click the "Connect" button to connect the Executor to the Manager.

6.7 Alchemi Console
The Console (Alchemi.Console.exe) is a grid administration and monitoring tool. It is located in the bin directory.
The 'Application's tab lets you monitor running applications. The 'Executors' tab provides information on Executors. The 'Users' tab lets you manage users Traditional Multiple-
Processor Paradigm vs. Grid Paradigm To clarify the concepts discussed so far, consider the following analogies between traditional multithreaded programming and grid programming:

<table>
<thead>
<tr>
<th>Multiprocessor Machine</th>
<th>Grid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>Alchemi</td>
</tr>
<tr>
<td>Processor</td>
<td>Executor</td>
</tr>
<tr>
<td>OS Services (High-Level) - System API</td>
<td>Alchemi API</td>
</tr>
<tr>
<td>OS Services (Low-Level)</td>
<td>Manager</td>
</tr>
<tr>
<td>Process</td>
<td>Process Grid Application</td>
</tr>
<tr>
<td>Thread</td>
<td>Grid Thread</td>
</tr>
</tbody>
</table>

### 7. Result & Conclusion
The Alchemi Grid consisting of Manager (Pentium IV Core-2-Duo, 2.4 GHz with 1GB RAM,) running Windows 2K3 Enterprise Server) and Five Executors (Pentium IV Core-2-Duo desktop machines, 2.4 GHz with 1GB RAM with 1GB physical memory running Windows XP).

#### 7.1 Test Application & Methodology
The application utilizes the Alchemi grid thread model. The test application is the computation of the value of Pi to n decimal digits. The algorithm used allows the computation of the p'th digit without knowing the previous digits. The test was performed for a range of workloads (calculating 100, 200, and 300, 400 & 500 digits of Pi), each with one to Five Executors enabled. The workload was sliced into a number of threads, each to calculate 20 digits of Pi, with the number of threads varying proportionally with the total number of digits to be calculated. Execution time was measured as the elapsed clock time for the test program to complete on the Owner node.

<table>
<thead>
<tr>
<th></th>
<th>Executor 1 (One Executor used)</th>
<th>Executor 2 (Two Executors used)</th>
<th>Executor 3 (Three Executors used)</th>
<th>Executor 4 (Four Executors used)</th>
<th>Executor 5 (Five Executors used)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pi Digits(100)</td>
<td>78</td>
<td>74</td>
<td>72</td>
<td>68</td>
<td>62</td>
</tr>
<tr>
<td>Pi Digits(200)</td>
<td>97</td>
<td>89</td>
<td>83</td>
<td>76</td>
<td>70</td>
</tr>
<tr>
<td>Pi Digits(300)</td>
<td>131</td>
<td>106</td>
<td>92</td>
<td>84</td>
<td>74</td>
</tr>
<tr>
<td>Pi Digits(400)</td>
<td>136</td>
<td>114</td>
<td>100</td>
<td>90</td>
<td>76</td>
</tr>
<tr>
<td>Pi Digits(500)</td>
<td>143</td>
<td>123</td>
<td>112</td>
<td>95</td>
<td>82</td>
</tr>
</tbody>
</table>

Table 7.1: Time taken by executor to complete task

### 8. Future Scope
We are building larger grid (More number of Computers will participate from different institutions) and the complex problems like weather forecasting, factorization will be tested on it.

### 9. References

[6] Grid Computing -Making the Global Infrastructure a Reality By FRAN BERMAN University of California, San Diego & San Diego Supercomputer Center, USA GEOFFREY FOX Community Grids Lab, Indiana University, USA TONY HEY Director e-Science Core Programme & University of Southampton, UK -- WILEY SERIES IN COMMUNICATIONS NETWORKING & DISTRIBUTED SYSTEMS.


[16] User Guide for Alchemi 1.0 Authors: Krishna Nadiminti, Akshay Luther, Rajkumar Buyya Grid Computing and Distributed Systems (GRIDS) Laboratory Dept. of Computer Science and Software Engineering The University of Melbourne, Australia