

## ALCHEMI FRAMEWORK - KHẢ NĂNG XÂY DỰNG MÔI TRƯỜNG VÀ PHÁT TRIỂN CÁC ỨNG DỤNG TÍNH TOÁN LƯỚI

*Nguyễn Thành Thủy\**

### TÓM TẮT

Từ tính toán lưới (TTL – Grid Computing) đến điện toán mây (Cloud Computing) ngày nay không còn là một giải pháp hàn lâm hay thử nghiệm. TTL cho phép ảo hóa các chức năng tính toán phân tán cũng như các nguồn xử lý, băng thông mạng và khả năng lưu trữ. TTL cho phép người sử dụng và các ứng dụng truy cập thông suốt vào các tính năng điện toán rộng lớn [1]. Với những tiến bộ quan trọng về phần mềm triển khai, người ta hy vọng TTL sẽ đem sức mạnh của một siêu máy tính tới tất cả người dùng PC đơn lẻ trên thế giới. Trong bài báo này, chúng tôi giới thiệu các khả năng xây dựng nền tảng cho môi trường TTL, các kỹ thuật phát triển ứng dụng và triển khai lưới tính toán trên nền NET-Base Alchemi Framework.

**Từ khóa:** alchemi framework, tính toán lưới, điện toán mây, dịch vụ lưới, dịch vụ web

### 1. Đặt vấn đề

Xây dựng một lưới tính toán có thể đơn giản như việc cho phép một số lượng nhỏ PC hoặc server hoặc mạng lưu trữ, tận dụng những khả năng chưa được khai thác hết. Từ một quy mô triển khai ban đầu nhỏ, người sử dụng có thể dần dần hoặc lập tức mở rộng lưới tùy theo nhu cầu của doanh nghiệp. Trong đó, cần có một hạ tầng cung cấp các công cụ mềm để xây dựng nền tảng cho lưới tính toán và các ứng dụng khác, dựa trên công nghệ Grid một cách dễ dàng. Nó bao gồm các module công cụ, mỗi công cụ định nghĩa một giao diện phục vụ cấp cao và cung cấp nhiều toán tử cấp thấp, có thể thực hiện được trên những môi trường khác nhau. [4]

Trong bài báo này, chúng tôi sẽ giới thiệu các khả năng xây dựng nền tảng cho môi trường TTL, các kỹ thuật phát triển ứng dụng và triển khai trên lưới tính toán trên nền .NET-Base Alchemi Framework.

### 2. Tổng quan về Alchemi Framework

Alchemi framework được phát triển bởi trường Đại học Melbourne (Australia). Alchemi là một .NET-based framework cung cấp các giải pháp để xây dựng môi trường lưới và phát triển các ứng dụng thực thi trên lưới. Alchemi framework cho phép xây dựng ứng dụng lưới trên nền tảng công nghệ .NET của Microsoft, hỗ trợ ngôn ngữ lập trình hướng đối tượng C#/VB.NET, trên hạ tầng mạng máy tính ngang hàng (peer-to-peer). Alchemi cho phép khai thác hiệu năng nhàn rỗi của các máy tính để bàn (desktop grid environment) chạy trên hệ điều hành Microsoft Windows. [3]

Alchemi framework cho phép xây dựng các ứng dụng lưới một cách linh hoạt, trên nền tảng hướng đối tượng, cung cấp thêm một mô hình tập tin dựa trên công việc (*file-based job model*). Hỗ trợ Cross-platform thông qua giao diện web, cung cấp khả năng tùy biến linh hoạt cho phép xử lý “chủ động” (dedicated) hay “bị động” (non-dedicated) ở các nút lưới [2].

Kiến trúc lớp của Alchemi cho môi trường tính toán lưới máy tính để bàn (*Desktop grid computing environment*) được hiển thị trong hình 1. Alchemi cho phép mô hình chủ - khách (*master - worker*) được tính toán song song, trong đó một thành phần trung tâm là đơn vị độc lập được thực hiện và quản lý song song. Trong Alchemi, đơn vị thực hiện song song được gọi là “luồng lưới” (*grid thread*) và có các hướng dẫn để được thực thi trên một nút lưới, được gọi là “Manager”.

### 2.1. Mô hình ứng dụng Lưới của Alchemi

Alchemi xử lý các yêu cầu chức năng và dữ liệu một cách song song. Cả hai đều được hỗ trợ bởi hai thành phần ứng dụng song song: mô hình luồng lưới (*grid thread model*) và mô hình lưới công việc (*grid job model*).

#### 2.1.1. Mô hình luồng lưới (*grid thread model*)

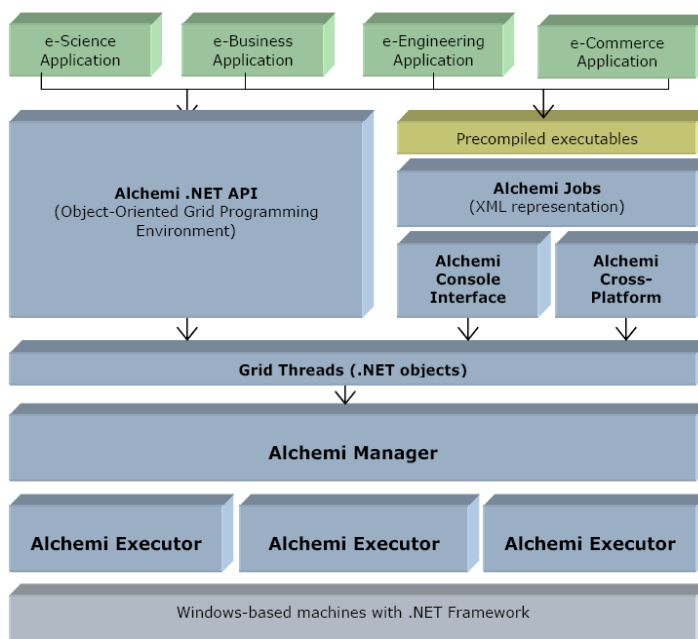
Mỗi luồng lưới là các đơn vị nguyên tử được thực hiện song song và độc lập, với nhiều luồng lưới là một ứng dụng lưới. Hai lớp trung tâm của Alchemi .NET API Là **GThread** và **GApplication**, đại diện cho một luồng lưới và ứng dụng lưới tương ứng.

Có hai thành phần cơ bản trong một ứng dụng lưới của Alchemi, đó là:

- "**Remote code**": thực hiện mã nguồn từ xa.
- "**Local code**": thực hiện mã nguồn trên máy cục bộ.

#### 2.1.2. Mô hình lưới công việc (*Grid Job Model*)

Việc triển khai lưới truyền thống đã được thực hiện ở một mức độ cao, và sự ảo hóa của các thiết bị, trong đó một đơn vị công việc nhỏ nhất thực hiện song song là một tiến trình. Trong mô hình này, một đơn vị làm việc thường được mô tả cụ thể bởi một câu lệnh, các tập dữ liệu đầu vào và các tập dữ liệu đầu ra. Trong Alchemi, một đơn vị làm việc được gọi là một “công việc” (job), với nhiều công việc tạo thành một "nhiệm

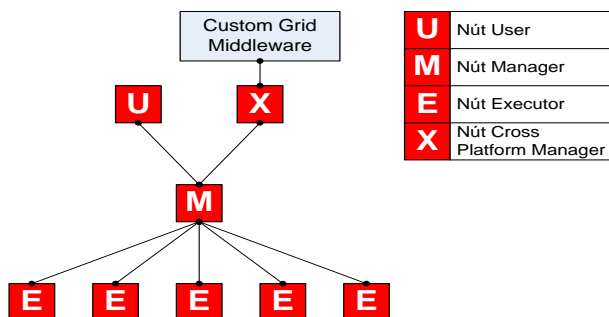


Hình 1. Kiến trúc phân lớp cho môi trường lưới máy tính để bàn của Alchemi

vụ" (task) [3].

**2.2. Các thành phần phân tán trong Alchemi**

Alchemi gồm có 4 thành phần phân tán chính, tham gia vào việc xây dựng lưới tính toán và thực hiện các ứng dụng lưới, bao gồm: **Manager (M)**, **Executor (E)**, **User (U)** & **Cross-Platform Manager (X)**.



**Hình 2.** Các thành phần phân tán của Alchemi

Thành phần **Manager** được triển khai ở máy chủ trung tâm, cung cấp các dịch vụ liên quan đến việc: tiếp nhận các yêu cầu TTL, thực thi các ứng dụng lưới và quản trị các luồng tính toán. Manager tiếp nhận các yêu cầu tính toán từ User, tổ chức lập lịch và điều phối các luồng xử lý đến các nút lưới (Executor) và trả kết quả về cho User.

Thành phần **Executor** được triển khai trên các nút lưới, có chức năng tiếp nhận các luồng lưới được yêu cầu từ Manager, xử lý và trả kết quả về lại cho Manager.

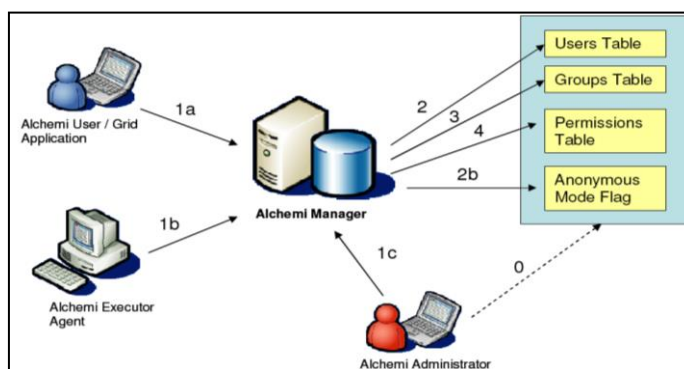
Thành phần **User** là nơi tiêu thụ các ứng dụng lưới của người dùng đầu cuối, các yêu cầu này sẽ được User gửi đi và nhận kết quả về từ Manager.

Thành phần **Cross-Platform Manager** cung cấp các dịch vụ để kiểm soát thành phần Manager thông qua giao diện web, bao gồm: kích hoạt dịch vụ lưới, quản trị việc thực hiện các công việc lưới,...

**2.3. Bảo mật trong Alchemi**

Bảo mật đóng một vai trò then chốt trong một môi trường không an toàn như Internet. Hai khía cạnh của bảo mật được giải quyết bởi Alchemi là: (a) cho phép người dùng thực hiện các hoạt động ủy quyền và (b) cho phép ủy quyền hoặc người dùng không được ủy quyền để đóng góp nguồn lực. [5]

Bằng cách sử dụng vai trò ủy quyền, chỉ cho phép người dùng thực hiện các hoạt động mà họ được phép làm. Mọi hoạt động



**Hình 3.** Cơ chế bảo mật trong

bảo mật nhạy cảm tại Manager được bảo hộ bằng cách này. Manager có thể được cấu hình để cho phép một Executor là vô danh hoặc ẩn danh.

Thông tin thực thi bảo mật của các nhóm và người dùng được duy trì trong cơ sở dữ liệu của Alchemi, được tổ chức trong các thực thể: **PRM**, **GRP** và **USR**. Mỗi hoạt

động về quản lý và liên kết đều tuân theo các quy định sau:

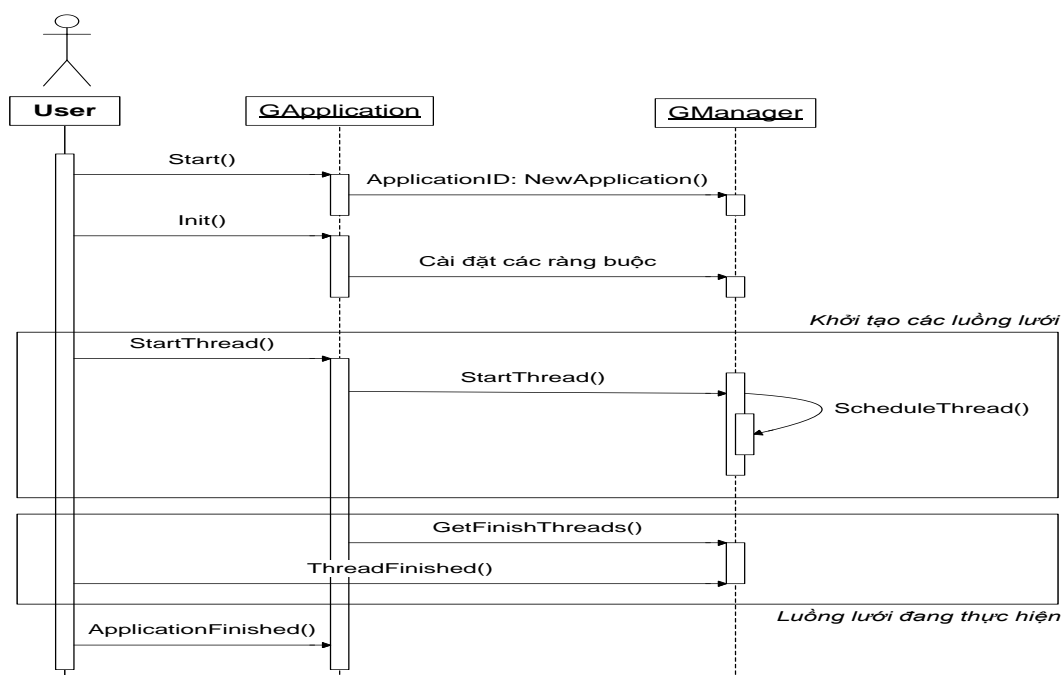
- **ExecuteThread:** Hoạt động liên quan đến thực hiện luồng, ví dụ như: nhận một luồng để thực thi, trả về kết quả.
- **ManageOwnApp:** Hoạt động liên quan đến quyền sở hữu của một ứng dụng cụ thể, ví dụ như tạo ra một ứng dụng, nhận về kết quả đã hoàn thành.
- **ManageAllApps:** Hoạt động liên quan đến quyền sở hữu của tất cả các ứng dụng trong hệ thống, ví dụ như: nhận về một danh sách tất cả các ứng dụng cùng với số liệu thống kê.
- **ManageUsers:** Hoạt động liên quan đến quản lý người dùng, ví dụ như việc thêm người dùng, thay đổi mật khẩu, thay đổi thành viên nhóm, ...

Khía cạnh thứ hai của bảo mật mà Alchemi cho phép, là bảo vệ Executer từ các mã độc hại. Bằng cách tạo ra một “hộp kín” là môi trường để Executer thực thi các luồng lưới. Môi trường này có thể được định nghĩa bởi người sử dụng.

### 2.3.1. Cài đặt chương trình

Một số phương thức chính được sử dụng trong chương trình thực nghiệm:

- **Init():** Phương thức khởi tạo dịch vụ lưới, thực hiện các kết nối vào lưới dịch vụ.
- **Start():** Phương thức kích hoạt các luồng lưới trên nút **Manager**.
- **App\_ThreadFinish():** Phương thức bắt sự kiện (Event) khi một luồng lưới được thực hiện hoàn thành. Ở đây, chúng ta có thể đặt các phương thức tương ứng thực hiện việc tiêu thụ các kết quả xử lý của luồng lưới.
- **App\_ApplicationFinish():** Phương thức bắt sự kiện khi một ứng dụng lưới được thực hiện hoàn thành. Ở đây, chúng ta có thể đặt các phương thức tương ứng thực hiện việc giải phóng tài nguyên lưới có huy động trong các luồng đã phát sinh.



Hình 4. Biểu đồ hoạt động chính của hệ thống

### 3. Xây dựng ứng dụng lưới trên nền Alchemi Framework

#### 3.1. Cài đặt và triển khai môi trường lưới

Bộ cài đặt và các tài liệu hướng dẫn cấu hình Alchemi được phân phối miễn phí tại trang <http://www.cloudbus.org/~alchemi/>. Việc cài đặt và cấu hình diễn ra khá đơn giản. Bộ cài đặt gồm 2 gói, gói **Alchemi.ManagerExecSetup.msi** dành cho **Manager** và **Alchemi.ExecutorExecSetup.msi** dành cho **Executor**.

#### 3.2. Phát triển ứng dụng trên lưới

##### 3.2.1. Lựa chọn bài toán:

Để đánh giá tính khả thi của Alchemi trong việc xây dựng ứng dụng lưới tính toán, chúng ta cần lựa chọn bài toán có độ phức tạp lớn đòi hỏi hiệu năng tính toán cao của hệ thống, nhưng không quá tốn kém về chi phí truyền thông. Bài toán được lựa chọn là sinh ngẫu nhiên n số nguyên tố (SNT) siêu mạnh 1024 Bit [6]. Để khai thác tối đa ưu điểm của lưới, thuật toán cần được song song hóa.

##### 3.2.2. Phương pháp thực nghiệm

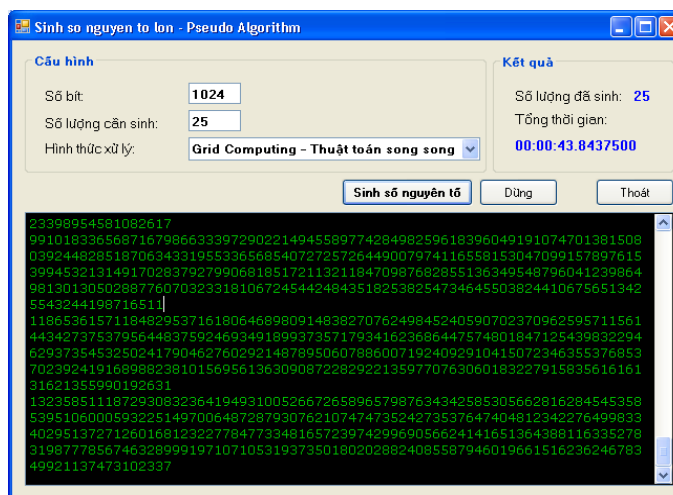
Đầu tiên chúng tôi cho thử nghiệm bài toán trên một PC, lần lượt cho thực hiện với thuật toán xử lý tuần tự và thuật toán đã được song song hóa. Sau đó triển khai trên lưới với số lượng số n tăng dần, ghi nhận kết quả làm cơ sở để đánh giá tính khả thi.

Trong đó, thuật toán sinh SNT siêu mạnh được song song hóa bằng chiến lược song song hóa kết quả. Tiến trình cha khởi tạo công việc kiểm tra số lượng SNT đã được sinh ra, khởi tạo các tiến trình con, mỗi tiến trình con thực hiện sinh SNT siêu mạnh và trả kết quả về cho tiến trình cha. Trong khi vẫn chưa đủ số n thì tiếp tục lặp lại công việc trên.

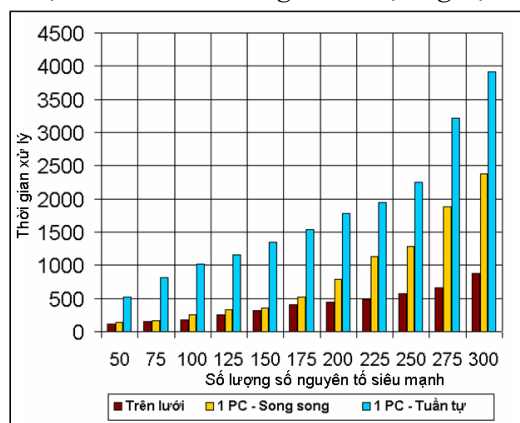
##### 3.2.3. Môi trường thực nghiệm

Tất cả PC được sử dụng trong thực nghiệm có cấu hình đồng bộ: CPU Intel Pentium Dual-Core 3.00 GHz, 2 MB Cache, 2 cores, 2 threads, DDR3 2GB/1333Hz. Được kết nối trên mô hình mạng LAN.

##### 3.2.4. Phân tích kết quả thực nghiệm



Hình 5. Giao diện chính của chương trình thực nghiệm



Hình 6. Biểu đồ tương quan giữa thời gian xử lý với số lượng SNT được sinh

Kết quả thực nghiệm được biểu diễn trong biểu đồ ở hình 6, phản ánh mối tương quan giữa số lượng SNT phát sinh và thời gian xử lý (*đơn vị giây*), lần lượt được xử lý với thuật toán tuần tự, thuật toán song song trên 1 PC và trên lưới tính toán.

Qua biểu đồ cho thấy, khi thực hiện với thuật toán tuần tự trên 1 PC, thời gian xử lý của hệ thống khi sinh SNT siêu mạnh tăng dần theo số lượng lệnh phát sinh. Thời gian trung bình để xử lý lệnh cũng tăng dần, năng lực xử lý của hệ thống giảm dần. Ta thấy xuất hiện tình trạng quá tải của hệ thống khi số lượng giao dịch tăng lên.

Hiện tượng trên cũng xuất hiện khi thực nghiệm với thuật toán song song trên 1 PC nhưng hiện tượng suy giảm nhanh của năng lực xử lý xuất hiện chậm hơn so với xử lý trên thuật toán tuần tự.

Khi sử dụng lưới tính toán để xử lý, thời gian xử lý của hệ thống giảm kéo theo thời gian trung bình để xử lý lệnh giảm rõ rệt, dấu hiệu quá tải chưa thấy xuất hiện trong hệ thống. Tuy nhiên, khi huy động càng nhiều nút lưới trong lưới tính toán thì mức độ cải thiện về thời gian xử lý vẫn không đáng kể, nguyên nhân ở đây là do chi phí đường truyền, chi phí xử lý các nghiệp vụ phân luồng và lập lịch,... của nút Manager cũng sẽ tăng khi có thêm nhiều nút lưới trong lưới tính toán.

#### 4. Kết luận

Những mô hình tính toán gần đây cũng tiếp cận đến cách khai thác phần mềm, khai thác nền tảng như một dịch vụ, giải pháp máy chủ ảo, ... và đã chuyển dần sang mô hình điện toán mây. Việc nghiên cứu và ứng dụng các mô hình tính toán mới hiện đang nằm trong chiến lược phát triển CNTT của nhiều quốc gia, trong đó có Việt Nam.

Trong bài báo này, chúng tôi đã giới thiệu công cụ Alchemi Framework, dễ dàng cho trong việc triển khai môi trường và phát triển các ứng dụng tính toán lưới. Alchemi không yêu cầu cấu hình của hệ thống quá phức tạp, hỗ trợ nền tảng .NET, ứng dụng lưới có thể được phát triển trên các ngôn ngữ C#/VB.NET, đặc biệt hoạt động tốt trên nền Microsoft Windows.

#### TÀI LIỆU THAM KHẢO

- [1] Ahmar Abbas, *Grid Computing: A Practical Guide to Technology and Applications*. Ian Foster, What Is The Grid? A Three Point Checklist, Gridtoday. July 2002.
- [2] Akshay Luther, Rajkumar Buyya, Rajiv Ranjan, and Srikumar Venugopal, *Alchemi: A .NET-based Grid Computing Framework and its Integration into Global Grids*.
- [3] Akshay Luther, Rajkumar Buyya, Rajiv Ranjan, and Srikumar Venugopal. *Peer-to-Peer Grid Computing and a .NET-based Alchemi Framework*.
- [4] Ian Foster, Carl Kesselman, Jeffrey Nick, Steve Tuecke, *Grid Services for Distributed System Integration, Computer*. June 2002.

- [5] Krishna Nadiminti, Akshay Luther, Rajkumar Buyya, *Alchemi: A .NET-based Enterprise Grid System and Framework*.
- [6] Nguyễn Văn Phương, *Vai trò của số nguyên tố siêu mạnh và việc sinh ra chúng*. Luận án tiến sĩ, 2008.

ALCHEMI FRAMEWORK – POSSIBILITY OF SETTING UP ENVIRONMENT  
AND DEVELOPING GRID COMPUTING APPLICATIONS

*Nguyen Thanh Thuy*

*University of economics, The University of Da Nang*

**ABSTRACT**

Grid Computing as well as Cloud Computing is not still an academic or pilot solution. Grid computing allows virtual ciphertext of distributed computing functions such as processing sources, network bandwidth and storage capacity. Users and applications are permitted to access enlarged computing functions smoothly. With advanced improvements of the deployed software, it is hoped that the power of super computers will be brought to everyone who uses PC. In this article, we will introduce possibilities of the foundation for Grid Computing environment, techniques for development of applications and deployment of Grid computing on NET-Base Alchemi Framework.

**Keywords:** alchemi framework, grid computing, cloud computing, grid service, web service.

\* Nguyễn Thành Thủy, Email: thuynt.it@gmail.com, Trường Đại học Kinh tế, ĐHQĐN