BỘ GIÁO DỤC VÀ ĐÀO TẠO
TR- ỜNG ĐẠI HỌC DÂN LẬP HẢI PHÒNG
-------o0o-------

TÌM HIỂU NGHIÊN CỨU MỘT SỐ BÀI TOÁN VỀ AN TOÀN THÔNG TIN TRONG TÍNH TOÁN LƯỚI

ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ Thông tin

Sinh viên thực hiện: Nguyễn Thị Trang
Giáo viên h- ớng dẫn: PGS TS. Trịnh Nhật Tiến
Mã số sinh viên: 111363

HẢI PHÒNG - 2011

MỤC LỤC

BẢNG CHỮ VIẾT TẮT

Từ viết tắt	Nghĩa Tiếng Anh	Chú giải

API	Application Programming Interface	Giao diện lập trình ứng dụng, thường là một tập các hàm giúp lập trình viên dễ dàng tương tác với dịch vụ hoặc hệ thống
DTP	Data Transfer Process	Tiến trình quản lý việc truy cập dữ liệu thực sự và truyền qua kênh dữ liệu trong kiến trúc GridFTP
GRAM	Grid Resource Allocation Management	Quản lý định vị tài nguyên lưới
FTP	File Transfer Protocol	Giao thức truyền tệp qua mạng
Globus XIO	Globus Xtensible Input/Output	Giao diện vào ra mức thấp nhất trong kiến trúc Globus
GridFTP	Grid File Transfer Protocol	GridFTP là giao thức mở rộng của FTP, tích hợp khả năng bảo mật lưới, truyền dữ liệu tốt hơn so với FTP
GSI	Grid Security Infrastructure	Hạ tầng an toàn thông tin lưới
HTTP	Hypertext Transfer Protocol	Giao thức truyền siêu văn bản
LFN	Logical File Name	Tên logic của thực thể dữ liệu
LRC	Local Replica Catalogue	Catalog định vị bản sao địa phương
PI	Protocol Interperter	Bộ thông dịch giao thức có nhiệm vụ quản lý các kênh điều khiển trong kiến trúc Grid FTP
RLI	Replica Location Index	Lưu các thông tin chỉ mục cho dịch vụ định vị bản sao
RLS	Replica Location Service	Dịch vụ định vị bản sao trong kiến trúc lưới dữ liệu Globus
RSL	Resource Specification Language	Ngôn ngữ đặc tả tài nguyên
SOAP	Simple Object Access Protocol	Giao thức truy cập đối tượng từ xa đơn giản
SSL	Secure Socket Layer	Giao thức bảo mật lưới
MDS	Monitoring and Discovery Service	Dịch vụ theo dõi và định dạng tài nguyên

LỜI CẢM ƠN

Để hoàn thành đồ án này, em đã nhận được rất nhiều sự giúp đỡ, chỉ bảo tận tình của các thầy cô. Xin gửi lời cảm ơn tới tất cả mọi người, đặc biệt xin chân thành cảm ơn:

Sự quan tâm giúp đỡ, chỉ bảo nhiệt tình của Thầy giáo Trịnh Nhật Tiến- Bộ môn Công nghệ thông tin trường Đại học Quốc Gia Hà Nội.
Sự giúp đỡ và tạo mọi điều kiện của các thầy cô trong Ban giám hiệu nhà trường nói chung và Bộ môn công nghệ thông tin nói riêng.
Một lần nữa xin chân thành cảm ơn.

Sinh viên

Nguyễn Thị Trang

GIỚI THIỆU

Trong vài năm trở lại đây tính toán mạng lưới đã phát triển mạnh mẽ, mở ra các giải pháp mới cho các ứng dụng đòi hỏi khả năng tính toán lớn. Grid computing có thể được sử dụng cho các bài toán nghiên cứu về sinh học, y học, vật lý, hoá học...cũng như các ứng dụng trong phân tích và đánh giá tài chính, khai thác dữ liệu và rất nhiều các loại ứng dụng khác.

Trong đồ án này, em xin trình bày một cách tổng quan về công nghệ Grid computing như: lợi ích, các thành phần, phạm vi ứng dụng của lưới tính toán. Trên cơ sở đó đi sâu vào tìm

hiểu về hệ thống bảo đảm an toàn thông tin và một số bài toán về an toàn thông tin trong tính toán lưới.

Chương1. TỔNG QUAN VỀ TÍNH TOÁN LƯỚI

1.1. KHÁI NIỆM TÍNH TOÁN LƯỚI

Ngày nay với sự phát triển vượt bậc của khoa học kỹ thuật và công nghệ đã xuất hiện những bài toán trong nhiều lĩnh vực đòi hỏi sức mạnh tính toán mà một máy tính riêng lẻ không thể đảm trách. Tính toán lưới ra đời nhằm tạo khả năng chia sẻ tài nguyên trên phạm vi toàn cầu, khả năng tận dụng các phần mềm cũng như tài nguyên vật lý phân tán cả về mặt địa lý.

Định nghĩa 1:

+ Lưới tính toán: là một hệ thống máy tính ảo, sử dụng máy tính rỗi tại nhiều nơi, trong các thời gian khác nhau.

+ Tính toán lưới: là việc tính toán trên **Lưới tính toán.**

***Định nghĩa của IBM*:**

Tính toán lưới là một môi trường tính toán ảo. Môi trường này cho phép bố trí song song, linh hoạt, chia sẻ, tuyển lựa, tập hợp các nguồn tài nguyên hỗn hợp về mặt địa lý, tùy theo mức độ sẵn sàng, hiệu suất, chi phí của các tài nguyên tính toán và yêu cầu về chất lượng dịch vụ của người sử dụng.

1.2. LỢI ÍCH CỦA TÍNH TOÁN LƯỚI

1.2.1. Khai thác tài nguyên nhàn rỗi

Một trong những lợi ích cơ bản của tính toán lưới là khả năng chạy ứng dụng trên một tài nguyên khác. Thống kê cho thấy, đối với các máy tính để bàn, trong một ngày làm việc thì chỉ có khoảng 5% thời gian là bận, còn lại là rỗi. Việc tận dụng khoảng thời gian rỗi này để chạy các ứng dụng khác là một việc làm rất hiệu quả và kinh tế.

1.2.2. Khả năng xử lý song song

Khả năng chạy ứng dụng song song là khả năng hấp dẫn nhất mà tính toán lưới mang lại. Lúc này, một công việc được chia thành nhiều công việc con, các công việc con này được thực hiện đồng thời trên các tài nguyên khác nhau của lưới. Do đó, thời gian chạy ứng dụng sẽ được rút ngắn nhiều lần.

Tuy nhiên, vấn đề không phải ứng dụng nào cũng có thể triển khai theo cách này được. Cần xem xét các yếu tố như khả năng song song hóa, sự trao đổi giữa các công việc con khi chạy để đánh giá xem một ứng dụng có thực sự hiệu quả khi được triển khai trên lưới hay không.

1.2.3. Sự cộng tác các tài nguyên ảo và tổ chức ảo

Sự hợp tác được thể hiện thông qua khái niệm tổ chức ảo – sự kết hợp nhiều tổ chức thực cùng mục tiêu. Thông qua mô hình tổ chức ảo, các tổ chức thực có thể chia sẻ tài nguyên

như dữ liệu, các thiết bị đặc biệt…Những tài nguyên này được "ảo hóa" để giữ chúng đồng bộ trong một hệ thông mạng lưới không đồng nhất. Các tài nguyên đó gọi là tài nguyên ảo.

1.2.4. Giúp truy nhập các tài nguyên khác

Ngoài tài nguyên tính toán và lưu trữ, lưới còn cung cấp các loại tài nguyên khác, chẳng hạn đường truyền mạng, các phần mềm đắt tiền. Ví dụ như nếu một người dùng muốn tăng thông lượng kết nối tới Internet để thực hiện khai phá dữ liệu, anh ta có thể tận dụng các kết nối Internet riêng biệt của các nút lưới khác để chạy bài toán trên.

1.2.5. Giúp cân bằng trong sử dụng tài nguyên

Lưới liên kết các tài nguyên từ nhiều máy khác nhau tạo thành một hệ thống duy nhất. Lưới có thể thực hiện cân bằng tài nguyên trong các chương trình bằng cách lập lịch làm việc cho các công việc. Chức năng này có ý nghĩa rất lớn trong việc xử lý các trường hợp quá tải về xử lý, tính toán trong một tổ chức. Chức năng cân bằng có thể được thực hiện theo 2 cách sau:

- Những điểm quá tải được đưa đến những máy rỗi trên mạng lưới.

- Nếu toàn mạng lưới đã bận, những công việc có độ ưu tiên thấp được tạm ngừng nhường cho những công việc khác có độ ưu tiên cao.

Một lợi ích khác khi dùng Grid là cân bằng tải. Khi một công việc liên lạc với một công việc khác, với Internet , hoặc các tài nguyên khác, Grid có thể lập lịch cho chúng để có thể giảm thiểu tối đa lưu lượng đường truyền cũng như khoảng cách truyền. Điều này giúp Grid có thể giảm thiểu tối đa lưu lượng đường truyền cũng như khoảng cách truyền. Điều này giúp Grid có thể giảm thiểu tắc nghẽn mạng.

1.2.6. Mang lại độ tin cậy

Khái niệm tin cậy trong tính toán lưới được thể hiện ở các khía cạnh sau:

- Trong lưới có những tài nguyên tính toán đắt tiền, cung cấp độ tin cậy cao cho những bài toán được thực hiện trên chúng

- Lưới cung cấp khả năng lập lịch lại, phân bổ lại công việc nếu có lỗi xảy ra

- Nếu cần, một công việc có thể được chạy đồng thời trên nhiều nút, cho nên việc xảy ra lỗi ở một nút sẽ không làm ảnh hưởng đến kết quả của công việc đó.

1.2.7. Phạm vi ứng dụng

Tính toán lưới thường được sử dụng để giải quyết các bài toán khoa học đòi hỏi khả năng tính toán và thông lượng cao như mô phỏng, thiết kế vi mạch, chia sẻ nội dung, truy nhập/thuê các phần mềm/dịch vụ từ xa. Hoặc các bài toán đòi hỏi dữ liệu lớn, thời gian thực, phục vụ theo yêu cầu và các bài toán tính toán cộng tác như thiết kế cộng tác, khai phá dữ liệu, giáo dục điện tử…

1.3. THÀNH PHẦN CỦA LƯỚI TÍNH TOÁN THEO MÔ HÌNH CHỨC NĂNG

Trong mô hình chức năng của lưới, có rất nhiều thành phần như: thành phần bảo vệ thông tin, thành phần môi giới, thành phần lập lịch, chức năng an ninh nút, thành phần quản lý tài nguyên, thành phần quản lý dữ liệu, thành phần giao thức, nhưng trong chương này em chỉ trình bày các thành phần cơ bản của nó.

1.3.1. Thành phần Bảo vệ thông tin

Bảo vệ thông tin luôn là một thành phần quan trọng trong bất kỳ một hệ thống tính toán nào, trong đó có môi trường lưới. Khi người sử dụng thực hiện công việc từ xa trên hệ thống khác, họ thường xuyên quan tâm tới việc liệu hệ thống đó có đảm bảo được rằng công việc và dữ liệu của họ không bị truy cập trái phép. Còn nhà cung cấp dịch vụ thì lại phải đảm bảo ứng dụng lưới không làm gián đoạn các ứng dụng đang chạy trên máy người dùng, không giao tiếp truy cập với các dữ liệu cá nhân.

1.3.2. Thành phần Quản lý tài nguyên lưới

1.3.2.1. Những thách thức trong quản lý tài nguyên lưới

1/. Xuất phát từ đặc trưng của tài nguyên lưới:

Lưới không giữ quyền điều khiển tuyệt đối đối với các tài nguyên, ta phải phát triển phương thức quản lý trên các vùng khác nhau và nguồn tài nguyên không đồng nhất.

Các tài nguyên lưới là không đồng nhất tại các tổ chức khác nhau, các chính sách quản lý tài nguyên quản lý và mục đích của người dùng tài nguyên lại khác nhau hoặc thậm chí có thể mâu thuẫn nhau. Hầu hết các ứng dụng yêu cầu sử dụng đồng thời nhiều tài nguyên ở nhiều nơi khác nhau để hoàn thành công việc.

Một thách thức khác đối với quản lý tài nguyên là vấn đề bảo vệ tài nguyên. Vì hệ thống lưới là phân tán cả về địa lý và tổ chức. Các tổ chức khác nhau có các chính sách bảo vệ khác nhau. Một lưới có đa dạng nguồn tài nguyên, mỗi tài nguyên lại yêu cầu các mức bảo vệ khác nhau.

2/. *Định vị tài nguyên lưới:*

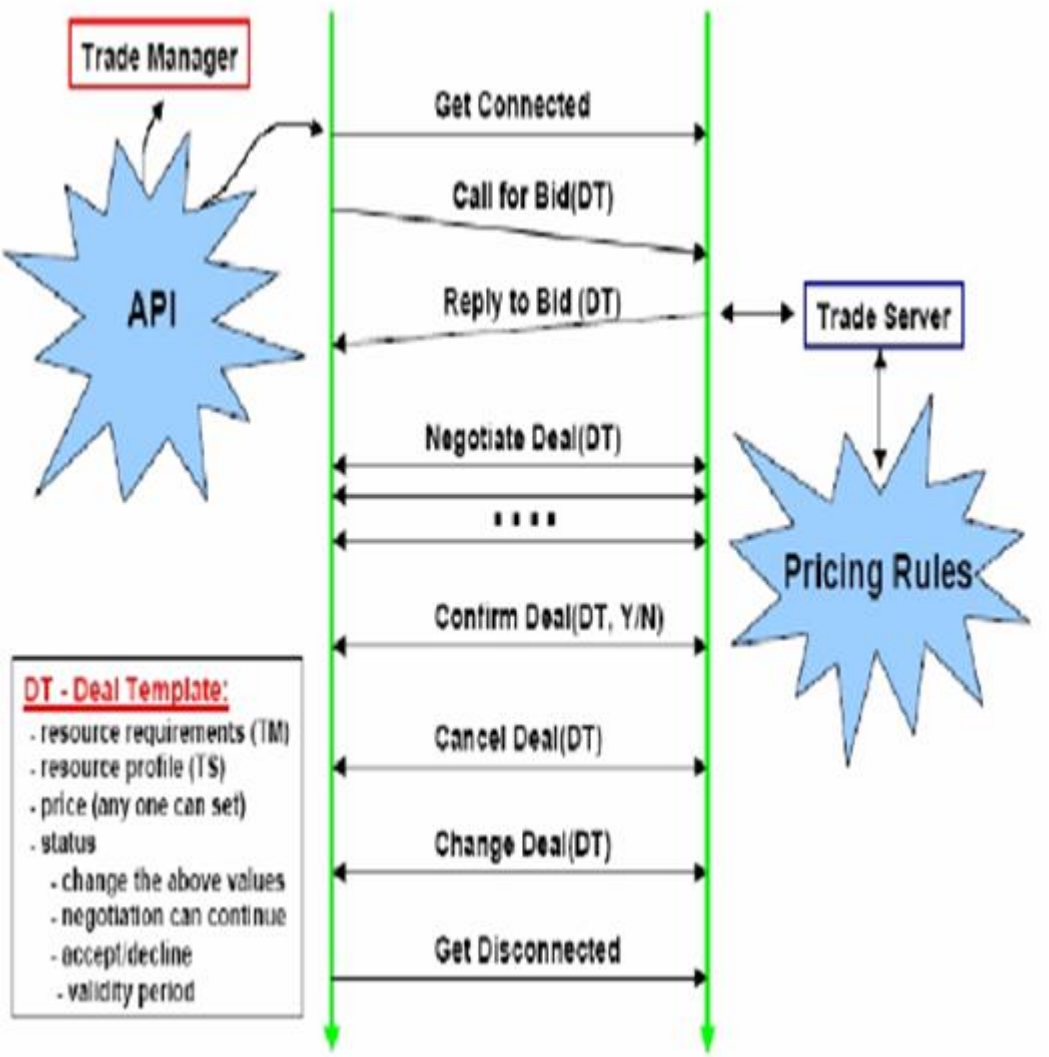
Khi có yêu cầu của người dùng, bộ phận quản lý tài nguyên sẽ tìm tài nguyên từ dịch vụ chỉ mục (Index Service) sau đó định vị tài nguyên đến một số nút cụ thể nào đó trong lưới và tại các nút này thì tài nguyên sẽ được lập lịch sử dụng. Khi một ứng dụng đang chạy, bộ phận quản lý tài nguyên cần theo dõi trạng thái tài nguyên và thông báo trở lại cho bộ lập lịch và hệ thống kế toán. Khi có 2 yêu cầu được đệ trình đến lưới cùng lúc thì cả 2 sẽ cùng được xử lý theo quy ước hoạt động của hàng đợi. Khi một ứng dụng yêu cầu sử dụng tài nguyên mà hiện tại tài nguyên đó đang phục vụ cho một ứng dụng khác thì nó sẽ được xếp vào hàng đợi cho đến khi tài nguyên đó được sử dụng xong và sẵn sàng phục vụ.

Môi trường lưới phân tán về địa lý và tài nguyên lưới là không đồng nhất, nên để định vị đúng tài nguyên, ta cần phải thiết kế một hệ thống quản lý tài nguyên phù hợp và phải chuyển sang hướng tiếp cận đa tầng và tổ chức tài nguyên phi tập trung.

3/. Vấn đề thương lượng tài nguyên lưới

Quá trình thương lượng tài nguyên lưới dựa trên các giao thức hay các luật trong kinh doanh để chuyển đổi các lệnh buôn bán giữa người sử dụng tài nguyên và các nhà cung cấp tài nguyên. Hình 1.1 minh họa các giao thức thương lượng mà cả hai phía mua và bán cần trong quá trình mặc cả.

Đầu tiên, phía khách hàng kết nối với nhà cung cấp. Sau khi nhận được giá tài nguyên, cả hai bên bán và mua sẽ tiến hành thương lượng. Khi thương lượng thành công, phía khách hàng sẽ yêu cầu ngừng kết nối và sử dụng tài nguyên đó.



Hình 1.1 Mô hình thương lượng tài nguyên lưới

1.3.2.2. Hệ quản trị tài nguyên GRAM

GRAM (Grid Resource Allocation Management) là dịch vụ được xây dựng trên cơ chế bảo mật GSI (Grid Security infrastructure), nó đóng vai trò là bộ phận quản lý, phân chia tài nguyên trong toàn bộ hệ thống tính toán lưới.

***Kiến trúc của GRAM**:*

1/. Kiến trúc bên ngoài:

Để có thể đệ trình một công việc lên một host, người dùng sẽ thông qua các API (Application Programming Interface) của GRAM Client để xác lập các thông tin về tài nguyên mà công việc cần đồng thời tạo ra tiến trình mới. Những thông tin này sẽ được gửi đến người quản lý cổng tương ứng. Người quản lý cổng sẽ xác thực những thông tin được gửi đến dựa vào cơ chế bảo mật GSI. Nếu tất cả đều hợp lệ, người quản lý cổng sẽ tạo ra một quản lý công việc để phục vụ cho công việc. Người quản lý công việc sẽ phân tích kịch bản RSL (Resource Specification Language) do người sử dụng gửi tới. Những kết quả phân tích được ngay lập tức được gửi tới các nguồn tài nguyên cục bộ và tiến hành thực thi công việc. Bên cạnh đó, quản lý công việc cũng sẽ tạo ra các tiến trình làm nhiệm vụ theo dõi và điều khiển công việc trong suốt quá trình xử lý.

Trong lúc công việc đang thực thi hay đã thực thi xong, các nguồn tài nguyên cục bộ sẽ phải thường xuyên cập nhật thông tin tài nguyên về cho MDS (Monitoring and Discovery Service). MDS sau đó sẽ hiển thị những thông tin này cho phép người dùng xem xét và lựa chọn nguồn tài nguyên nào thích hợp với công việc của mình.

2/. Kiến trúc bên trong:

Để có thể thực thi một công việc từ xa, người quản lý cổng GRAM phải được chạy trên một máy tính từ xa, lắng nghe ở một cổng được quy định trước, công việc sẽ được thực thi trên máy tính từ xa đó. Việc thực thi bắt đầu khi ứng dụng người dùng chạy trên máy cục bộ gửi yêu cầu đến máy tính từ xa. Yêu cầu đó sẽ mang các thông tin về lệnh thực thi, luồng vào, luồng xuất cũng như các thông tin về tên và cổng giao tiếp của máy tính từ xa. Yêu cầu công việc sẽ được xử lý bởi người quản lý cổng GRAM, từ đó nó sẽ tạo ra một quản lý công việc tương ứng mà công việc yêu cầu. Lúc đó, quản lý công việc sẽ theo dõi tình trạng thực thi công việc và chịu trách nhiệm thông báo thông tin của công việc cho người sử dụng.

1.3.3. Thành phần Quản lý dữ liệu

Quản lý dữ liệu là một phần quan trọng trong tính toán lưới nó cho phép truy nhập tài nguyên trên lưới với khối lượng lớn hàng giga-bytes thậm chí hàng tera- bytes dữ liệu. Quản lý dữ liệu phải đảm bảo được tính an toàn và ổn định trong quá trình di chuyển dữ liệu giữa các nút trong mạng lưới để hỗ trợ quá trình thực thi các công việc trong hệ thống tính toán lưới.

1.3.3.1. Giao thức truyền tập tin mạng lưới GridFTP

GridFTP là giao thức truyền tập tin giống như FTP hay truyền dữ liệu như HTTP. Đây là giao thức có hiệu năng cao, an toàn và đáng tin cậy nhất trên mạng Internet hiện nay. GridFTP được các nhà chuyên môn đánh giá cao vì nó cung cấp các tính năng đặc trưng phù hợp với kiến trúc mạng lưới như:

+ Bảo mật theo chuẩn GSI trên các kênh điều khiển và kênh truyền dữ liệu.

+ Tạo lập và quản lý các kênh truyền dữ liệu song song, cho phép tăng tốc độ truyền dữ liệu tới mức kỷ lục.

+ Trao đổi từng phần tập tin dữ liệu, đặc biệt hiệu quả với các tập tin dữ liệu có dung lượng cực kỳ lớn.

+ Trao đổi dữ liệu với sự tham gia của phía thứ ba. Đây là nghi thức cho phép chuyển tập tin trực tiếp từ máy chủ tới máy chủ khi kênh điều khiển nằm trên máy chủ thứ ba.

+ Xác thực các kênh truyền dữ liệu.

+ Tái sử dụng các kênh truyền dữ liệu và dẫn truyền các lệnh điều khiển.

- ***Mở rộng từ FTP***

GridFTP bao gồm một số chức năng trong giao thức FTP mở rộng và đã được chuẩn hóa, nhưng ít được cài đặt trong các hệ thống hiện tại. Các chức năng khác là các chức năng mới so với FTP như:

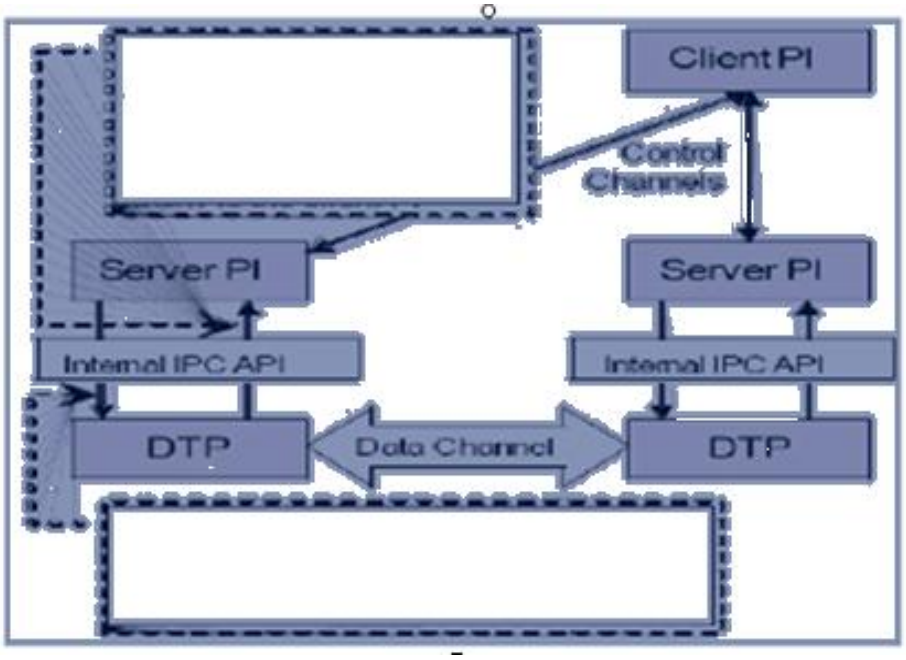
- Điều khiển bởi đối tác thứ ba.

- Truyền dữ liệu song song; phân đoạn và từng phần.

- Tự động thương lượng vùng đệm TCP.

- Truyền dữ liệu tin cậy và có khả năng khởi động lại.

- ***Kiến trúc của dịch vụ GridFTP***



Hình 1.2. Kiến trúc của dịch vụ GridFTP

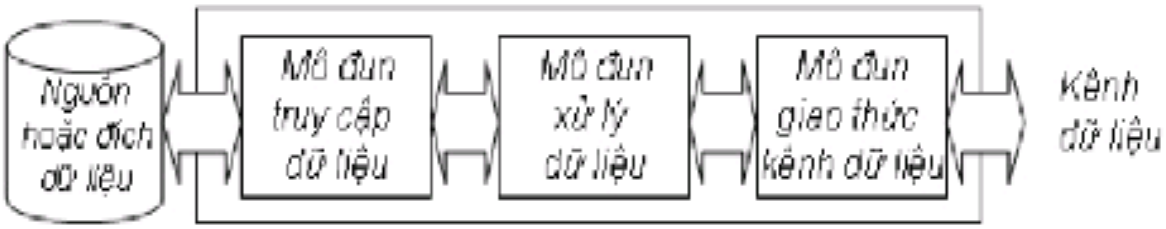
+ Bộ thông dịch giao thức PI:

Server PI có nhiệm vụ quản lý kênh điều khiển, trao đổi thông tin với máy khách qua kênh này. Để máy khách liên lạc với máy chủ GridFTP, server PI phải chạy như một chương trình thường trú, luôn lắng nghe ở cổng nào đó. Hoặc một dịch vụ khác của hệ thống phải lắng nghe trên cổng này, khi nhận được yêu cầu thì sẽ chuyển lời gọi tới Server PI. Tiếp đó, Client PI và Server PI "nói chuyện" với nhau qua giao thức đã định sẵn.

Trong suốt quá trình truyền thông, Server PI chỉ quan tâm tới việc xây dựng mô tả cho quá trình truyền dữ liệu. Thời điểm này, nó không liên hệ với DTP (Data Transfer Process) hoặc DTP có thể chưa chạy. Khi nhận lệnh yêu cầu hoạt động liên quan tới DTP, Server PI gửi bản mô tả quá trình truyền cho DTP. DTP tự thực hiện truyền dựa vào bản mô tả này. Khi bản mô tả được gửi đi, Server PI chỉ đóng vai trò là tầng chuyển tiếp các thông tin trạng thái.

+ Tiến trình truyền dữ liệu DTP:

Bản thân DTP được cấu tạo bởi ba môđun kết hợp như sau:



Hình 1.3. Đường ống truyền dữ liệu DTP

1/. Mô-đun truy cập dữ liệu: chịu trách nhiệm đọc/ghi dữ liệu tới nguồn/ đích. Giao diện truy cập gồm các lệnh gửi, nhận, tạo, xóa, đổi tên, tính tổng, kiểm tra.

2/. Mô-đun xử lý dữ liệu: xử lý dữ liệu phía máy chủ: nén, co dãn, ghép nối các tệp. Hiện tại chức năng xử lý dữ liệu được cài đặt cùng môđun truy cập dữ liệu.

3/. Mô-đun giao thức kênh dữ liệu: đảm nhiệm việc xử lý kênh dữ liệu, gồm các thao tác nạp/gửi dữ liệu. Một máy chủ có thể hỗ trợ nhiều kênh truyền dữ liệu.

- ***Bảo mật trong GridFTP***

GridFTP cung cấp việc chứng thực an toàn kênh điều khiển, đảm bảo tính toàn vẹn và bí mật cho kênh dữ liệu. Cơ chế bảo mật của nó xây dựng trên nền GSI. Phiên làm việc được thiết lập khi máy khách khởi tạo kết nối TCP tới cổng mà máy chủ GridFTP server đang lắng nghe. Đầu tiên diễn ra quá trình chứng thực. Đây là quá trình bắt tay ba bước. Máy khách trình một giấy ủy nhiệm, giấy này chứa thông tin về người dùng đại diện cho máy khách gồm định danh, khóa công khai, tên nhà thẩm quyền… Máy chủ cũng phải đưa ra một giấy chứng nhận riêng được cấp bởi nhà thẩm quyền mà máy khách tin tưởng.

Nếu quá trình kiểm tra thông tin trên các giấy chứng nhận này thất bại, liên kết không được thiết lập. Ngược lại, giai đoạn xác định thẩm quyền diễn ra: xác định quyền hạn truy cập của máy khách đối với dữ liệu trên máy chủ. Điều này được thực hiện bằng cách ánh xạ máy khách với một người dùng địa phương trên máy chủ. Quyền truy cập của người dùng địa phương sẽ tương đương với quyền truy nhập của máy khách. Thông tin ánh xạ được lưu trên máy chủ

trong một tệp grid- mapfile. Nếu chưa có thông tin ánh xạ, tiến trình truyền dữ liệu không được hoàn thành. Mặc định, kênh điều khiển được mã hóa để bảo đảm tính toàn vẹn.

- ***Cài đặt dịch vụ GridFTP***

Cài GridFTP trên nút lưới cung cấp dịch vụ. Nút này được gọi là máy chủ GridFTP. Cài GridFTP Client trên máy khách, thực hiện gửi yêu cầu tới máy chủ GridFTP để truy xuất dữ liệu.

Để thực hiện chức năng truyền tệp điều khiển bởi đối tác thứ ba, hai nút lưới tham gia quá trình truyền phải được cài đặt GridFTP Server.

1.3.3.2. Dịch vụ định vị bản sao RLS

Mục đích tạo bản sao là để làm giảm trễ truy cập, tăng tính địa phương của dữ liệu, tăng hiệu năng, khả năng mở rộng, và tính chịu lỗi của các ứng dụng phân tán. Hệ thống sử dụng bản sao cần có kỹ thuật xác định vị trí bản sao.

- ***Yêu cầu đối với một dịch vụ định vị bản sao***

RLS phải thỏa mãn các yêu cầu sau:

- Bản sao có tính chỉ đọc: RLS chỉ quản lý tệp không thay đổi hoặc thay đổi không thường xuyên, được định danh duy nhất dưới các phiên bản khác nhau.

- Phạm vi sử dụng: hệ thống phải có khả năng trải rộng trên hàng trăm miền, quản lý khoảng 50 triệu tệp lôgic và 500 triệu bản sao vật lý.

- Hiệu năng: hệ thống phải có khả năng hỗ trợ khoảng 1000 truy vấn và 200 lần cập nhật trên một giây. Thời gian hồi đáp trung bình phải ít hơn 10 miligiây, và thời gian hồi đáp truy vấn trung bình không vượt quá 5 giây.

- Bảo mật: RLS quan tâm nhiều nhất tới bảo vệ tính riêng tư và toàn vẹn của thông tin tồn tại và vị trí dữ liệu.

- Tính nhất quán: RLS không hỗ trợ khung nhìn nhất quán hoàn toàn đối với các bản sao.

- Tính tin cậy: lỗi xảy ra ở một miền không ảnh hưởng tới toàn bộ hoạt động của hệ thống.

- ***Kiến trúc của dịch vụ định vị bản sao***

Kiến trúc của dịch vụ quản lý bản sao phải đảm bảo được yêu cầu thực thi trên môi trường phân tán cao. Trong kiến trúc RLS, máy chủ định vị bản sao cục bộ cho từng miền được gọi là LRC (Local Replica Catalog). Máy chủ thực hiện nhiệm vụ đánh chỉ mục các LRC. Giao diện truy xuất của người sử dụng được gọi là RLI (Replica Location Index). Thông qua RLI, người sử dụng có thể tìm đến các LRC một cách dễ dàng. LRC phục vụ người dùng cục bộ trong tổ chức, còn RLI phục vụ người sử dụng trên phạm vi toàn bộ lưới. Như vậy, trên phạm vi toàn lưới dữ liệu, dịch vụ RLS được triển khai dưới dạng một tập các LRC phân tán tại site địa phương và một số RLI đánh chỉ mục cho các LRC.

+ Kho định vị bản sao cục bộ LRC:

LRC lưu giữ thông tin về các bản sao của một tổ chức cụ thể. LRC có một số chức năng:

- Về nội dung: lưu trữ ánh xạ giữa tên tệp lôgic bất kỳ với tên tệp vật lý.

-Về truy vấn: đáp ứng được các truy vấn: Cho một LFN, tìm tập các PFN tương ứng với LFN đó.

-Về tính toàn vẹn cục bộ: quản lý tính toàn vẹn giữa nội dung của tên logic với nội dung thực sự được lưu trên các hệ thống lưu trữ.

- Về bảo mật: thông tin trong LRC có thể liên quan đến điều khiển truy cập, vì thế hỗ trợ kỹ thuật chứng thực và xác nhận khi xử lý yêu cầu từ xa.

- Về sự lan truyền trạng thái: LRC thường xuyên gửi thông tin trạng thái- thông tin về sự thay đổi các ánh xạ tới RLI, bằng cách sử dụng thuật toán lan truyền trạng thái.

+ Chỉ mục định vị bản sao RLI:

LRC chỉ lưu trữ thông tin định vị bản sao tại các tổ chức, chỉ phục vụ người sử dụng trong phạm vi tổ chức đó. Nó không hỗ trợ người dùng truy vấn nhiều tổ chức cùng một lúc. Thông tin chỉ mục trong dịch vụ định vị bản sao được lưu dưới dạng một tập các RLI, mỗi RLI bao gồm tập bản ghi gồm hai trường (LFN, con trỏ tới LRC). RLI có thể đánh chỉ mục cho RLI khác.

Dựa trên kỹ thuật dư thừa, phân đoạn, và trạng thái mềm, có thể chỉ ra các yêu cầu đối với một nút chỉ mục định vị bản sao toàn cục RLI như sau:

- Truy cập từ xa an toàn: RLI phải hỗ trợ chứng thực, xác nhận, tính toàn vẹn, tính tin cậy, và phải triển khai quyền điều khiển truy cập cục bộ trên thông tin mà nó quản lý.

- Lan truyền trạng thái: RLI phải có khả năng nhận thông tin mô tả trạng thái do các LRC gửi đến định kỳ.

- Truy vấn: RLI phải trả lời truy vấn tới bản sao của một LFN cụ thể bằng cách trả về vị trí vật lý của LFN đó hoặc thông báo rằng LFN không nằm trong chỉ mục hiện thời, trong trường hợp không tìm thấy.

- Trạng thái mềm: RLI phải ấn định thời gian hết hạn đối với thông tin lưu trữ trong chỉ mục. Nếu một mục gắn liền với một LRC không nhận được thông tin trạng thái cập nhật từ LRC trong khoảng thời gian ấn định, RLI phải loại bỏ mục đó.

- Phục hồi khi lỗi xảy ra: RLI không được phép chứa thông tin trạng thái bền vững về các bản sao. Nó phải khôi phục nội dung sau sự cố chỉ bằng cách sử dụng cập nhật trạng thái động từ các LRC.

➢ ***Các tham số đặc trưng của kiến trúc RLS***

Để đặc tả một phạm vi rộng lớn kiến trúc của RLS, người ta dùng bộ sáu tham số (G, PL, P R, R, S, C). Bốn tham số đầu tiên (G, PL, P R, R) mô tả tính phân tán của thông tin bản sao. Hai tham số sau định nghĩa cách thông tin được gửi từ LRC đến RLI.

G: Số lượng RLI trong hệ thống.

PL: Đặc trưng cho kiểu phân nhóm tên tệp lôgic trong RLI.

PR: Đặc trưng cho kiểu phân nhóm không gian tên LRC.

R: Nói đến mức độ dư thừa trong việc đánh chỉ mục đối với mỗi tên tệp logic LFN.

S: Tần suất và cách thức cập nhật thông tin từ LRC đến RLI.
C: Phương pháp nén thông tin trao đổi giữa LRC và RLI.

1.3.4. Thành phần Lập lịch trong lưới tính toán

Sau khi xác định được tài nguyên cần thiết ta phải lập lịch trình các công việc được thực thi. Nếu các công việc là hoàn toàn độc lập thì có thể không cần bộ lập lịch. Nhưng thường thì ứng dụng đòi hỏi cần phải dự trữ tài nguyên nào đó, hoặc các công việc cần giao tiếp với nhau. Do đó, cần có bộ lập lịch để phối hợp các công việc.

Lập lịch trong lưới bao gồm 3 giai đoạn chính:

+ Khám phá tài nguyên và đưa ra danh sách tài nguyên có thể sử dụng được.

+ Lựa chọn tài nguyên phù hợp nhất đối với yêu cầu công việc.

+ Thực thi công việc.

- ***Giai đoạn 1: Khai phá tài nguyên***

Xác định xem tài nguyên nào khả dụng đối với người dùng hiện tại.

- Bước 1: Tìm các tài nguyên khả dụng: xác định tập tài nguyên mà người dùng có đủ thẩm quyền truy nhập tới.

- Bước 2: Xác định yêu cầu ứng dụng: người dùng phải định ra một tập các yêu cầu tối thiểu để thực hiện công việc để lọc các tài nguyên khả dụng.

- Bước 3: Loại bỏ những tài nguyên không đáp ứng được yêu cầu tối thiểu của công việc căn cứ vào danh sách các tài nguyên mà người dùng có quyền sử dụng và căn cứ vào kết quả phân tích yêu cầu công việc ở bước hai, ta loại bỏ tất cả những tài nguyên không đáp ứng được những yêu cầu tối thiểu của công việc. Đến cuối bước này người sử dụng sẽ có trong tay một tập các tài nguyên có thể dùng để triển khai công việc.

- ***Giai đoạn 2: Lựa chọn tài nguyên.***

Tiến hành thu thập các thông tin liên quan tới các yêu cầu còn lại của công việc và lựa chọn ra tài nguyên thích hợp nhất để thực thi công việc.

- Bước 1: Thu thập thông tin động: xác định xem thông tin nào sẵn có và người dùng có thể truy nhập đến nó như thế nào.

- Bước 2: Lựa chọn tài nguyên: sau khi đã có đầy đủ thông tin về tài nguyên người dùng sẽ lựa chọn những tài nguyên phù hợp nhất cho yêu cầu và mục đích của họ. Bước này thường do bộ lập lịch và quản lý tài nguyên thay mặt người dùng đảm nhận tự động bằng cách giải bài toán tối ưu.

- ***Giai đoạn 3: Thực thi công việc.***

Tiến hành các bước để thực thi công việc trên tài nguyên đã chọn, giám sát trạng thái công việc và gửi kết quả lại cho người sử dụng.

- Bước 1: Đặt trước tài nguyên (tùy chọn) để có thể sử dụng tốt nhất một hệ thống nào đó, một phần hoặc toàn bộ tài nguyên phải được đặt trước.

- Bước 2: Đệ trình công việc: sau khi đã chọn được tài nguyên ứng dụng, công việc cần phải được đệ trình lên tài nguyên đó để thực hiện bằng cách chạy một dòng lệnh đơn hoặc chạy một dãy các kịch …

- Bước 3: Các công việc chuẩn bị: trong bước này phía người dùng sẽ làm các công việc cần thiết để ứng dụng có thể chạy được. Ví dụ: dùng GridFTP để chuyển các file dữ liệu cần thiết đến địa điểm nơi công việc sẽ chạy.

- Bước 4: Theo dõi tiến độ: tùy thuộc vào ứng dụng và thời gian chạy của nó mà người

dùng có thể muốn theo dõi tiến độ và có thể sẽ thay đổi ý định của họ về việc công việc sẽ được thực hiện ở đâu và như thế nào.

- Bước 5: Hoàn thành công việc: khi công việc kết thúc thì cần phải báo cho người sử dụng bằng một hình thức nào đó.

- Bước 6: Dọn dẹp và kết thúc: sau khi một công việc đã được thực hiện xong, kết quả công việc phải được gửi lại cho người đệ trình, đồng thời các file tạm thời cũng phải được xóa đi.

1.3.5. Cổng lưới tính toán (Grid Portal)

Hệ thống tính toán lưới chỉ cung cấp cho người sử dụng một tập hợp các dịch vụ chung và khả năng khai thác các nguồn tài nguyên phân tán. Nó không cung cấp các thành phần giao diện thân thiện phục vụ người sử dụng. Vì vậy, đòi hỏi cần phải có một công cụ cung cấp các thành phần giao diện phục vụ người sử dụng. Trước thực tế đó, một cổng giao tiếp hệ thống Gird với tên gọi Grid Portal ra đời.

Grid Portal: là cổng kết nối dịch vụ giữa người dùng và nhà cung cấp dịch vụ, được phát triển như một phần mềm trên mạng Intenet để cung cấp các chức năng cần thiết theo hướng người dùng. Việc sử dụng công nghệ Portal cho phép tạo môi trường làm việc riêng biệt cho từng người dùng, đồng thời tách biệt các chức năng dịch vụ riêng biệt từ phía máy chủ và tái sử dụng các thành phần chức năng của Web.

Grid Portal được hình thành từ hai khái niệm cổng (portal) khác nhau: Cổng chuyên chưng dụng (Application Specific Portal) cung cấp một tập con các thao tác truy cập Grid chuyên biệt từ bên trong một ứng dụng, từ các miền đặc biệt. Và Cổng chuyên cho người dùng (User

Specific Portal) cung cấp các dịch vụ riêng liên quan đến các site phục vụ cho một tác vụ truyền thông nào.

Để triển khai công nghệ GridPortal, chúng ta có thể sử dụng công cụ phát triển GPDK (Grid Poratl Development Kits).

1.3.6. Thành phần Giám sát lưới

Trong môi trường lưới, nhu cầu giám sát các tài nguyên là rất cần thiết. Các tài nguyên của lưới luôn ở trạng thái động, chúng có thể gia nhập vào lưới rồi sau đó rút ra khỏi lưới vào bất kì thời điểm nào. Người dùng phải có khả năng tìm kiếm những tài nguyên mong muốn và giám sát các tài nguyên đó. Ngoài vai trò cung cấp thông tin cho người dùng, hệ thống giám sát lưới còn đóng vai trò quan trọng trong các hoạt động lập lịch, nhân bản dữ liệu, phân tích hiệu năng, xây dựng ứng dụng tự điều chỉnh...

1.3.6.1. Quy trình giám sát

Quy trình giám sát các hệ phân tán thường bao gồm bốn bước như sau:

1/. Sinh các sự kiện: bộ cảm biến tiến hành đo đạc trên các thực thể và mã hóa kết quả thu được theo một lược đồ cho trước.

2/. Xử lí các sự kiện: các sự kiện được xử lí theo từng ứng dụng cụ thể.

3/. Phân phối các sự kiện: các sự kiện được chuyển đến các bên quan tâm.

4/. Trình diễn các sự kiện: các sự kiện được xử lí để đạt tới mức trừu tượng cao, đủ để người dùng rút ra được kết luận về trạng thái của hệ thống. Giai đoạn này thường được thực hiện bởi một ứng dụng đồ họa, hiển thị dữ liệu tức thời theo thời gian thực hoặc lấy dữ liệu từ các kho lưu trữ và hiển thị.

1.3.6.2. Yêu cầu đối với một hệ thống giám sát lưới

Một hệ thống giám sát lưới phải thỏa mãn được những yêu cầu sau đây:

- Khả năng mở rộng: phải hoạt động tốt khi số lượng tài nguyên và người dùng tăng.

- Độ trễ xử lý nhỏ: trong lưới, các sự kiện liên tục được sinh ra với tốc độ cao và số lượng lớn, đồng thời để tránh tình trạng dữ liệu bị lạc hậu thì hệ thống giám sát phải có tốc độ xử lí dữ liệu cao nhằm đạt được độ trễ nhỏ nhất.

-Ít xâm phạm đến các tài nguyên: thao tác đo đạc diễn ra thường xuyên sẽ tiêu tốn đáng kể các tài nguyên. Hệ thống giám sát phải giữ được mức tiêu thụ tài nguyên của mình ở mức chấp nhận được.

- Hỗ trợ nhiều mô hình truyền dữ liệu: thông tin giám sát bao gồm các sự kiện tĩnh và các sự kiện động nên nó đòi hỏi các chính sách đo đạc khác nhau như đo định kì hay đo mỗi khi có yêu cầu.

- Khả chuyển: các tài nguyên trong lưới là rất không đồng nhất, bởi vậy các phần phần hệ thống giám sát phải có tính khả chuyển cao.

- Bảo mật: hệ thống giám sát phải hỗ trợ các dịch vụ bảo mật như điều khiển truy nhập, chứng thực, vận chuyển an toàn các thông tin giám sát.

- Khả năng đồng bộ hóa cao: bên nhận cần phải biết độ mới của một sự kiện do đó hệ thống giám sát phải có khả năng đồng bộ hóa cao giữa các thành phần.

1.3.6.3. Phân loại các hệ thống giám sát lưới

Các hệ thống giám sát được chia thành bốn mức như sau:

- Mức 0 (Level 0): các sự kiện được chuyển trực tiếp từ bộ cảm biến tới bộ tiêu thụ theo một trong hai chế độ online hoặc offline. Ở chế độ online, các kết quả đo đạc thường được truy nhập tới thông qua một giao diện web. Ở chế độ offline, kết quả đo được bộ cảm biến ghi vào kho lưu trữ và sau đó được bộ tiêu thụ lấy ra.

- Mức 1 (Level 1): trong các hệ thống loại này, các bộ cảm biến được xây dựng riêng và nằm trên cùng một máy với các bộ sinh, hoặc chúng được tích hợp vào trong các bộ sinh. Trong cả hai trường hợp, các sự kiện được truy nhập thông qua các API của bộ sinh.

- Mức 2 (Level 2): so với các hệ thống mức 1, các hệ thống mức 2 có thêm các thành phần trung gian. Các chức năng được phân bố trên cả bộ sinh và thành phần trung gian (có thể nằm trên máy khác) thay vì chỉ nằm trên một bộ sinh duy nhất.

-Mức 3 (Level 3): các hệ thống ở mức này có tính linh hoạt cao nhờ các thành phần trung gian được tổ chức theo cấu trúc phân cấp. Mỗi thành phần trung gian thu thập và xử lí các sự kiện từ các thành phần trung gian hay bộ sinh nằm ở mức thấp hơn và sau đó gửi chúng lên các thành

phần trung gian ở mức cao hơn. Các hệ thống mức 3 rất thích hợp cho môi trường lưới. Một hệ thống tiêu biểu thuộc loại này là Globus MDS.

1.4. CÁC THÀNH PHẦN CỦA LƯỚI TÍNH TOÁN THEO MÔ HÌNH VẬT LÝ

Các thành phần của lưới theo mô hình vật lý bao gồm:

1.4.1. Thành phần mạng (Networks)

Mạng đóng vai trò là cơ sở hạ tầng để truyền số liệu và các thông tin giám sát công việc giữa các điểm trong mạng lưới. Băng thông mạng là một thuộc tính rất quan trọng liên quan đến hiệu suất lưới.

1.4.2. Thành phần tính toán (Computation)

Cấp bởi các bộ xử lý trong lưới, chúng đa dạng về tốc độ, kiến trúc, nền tảng phần mềm và lưu trữ.

1.4.3. Thành phần lưu trữ (Storage)

Dữ liệu có thể được lưu trữ phân tán trên nhiều thiết bộ xử lý hoặc một mạng LAN. Mỗi bộ xử lý thường cung cấp một dung lượng lưu trữ nhất định. Hệ thống file thường được dùng là NFS, DFS hoặc GPFS.

1.4.4. Phần mềm và bản quyền (Software and License)

Về phương diện phần mềm trong môi trường tính toán lưới thì mức độ ổn định của ứng dụng phần mềm và bản quyền phần mềm là hai vấn đề cần được quan tâm nhất.

1.4.5. Các thiết bị đặc biệt

Một vài nút trên lưới có thể có những thiết bị đặc biệt, chẳng hạn các thiết bị quân sự, y tế, hay các thiết bị chuyên dụng khác.

1.5. HỆ THỐNG ĐẢM BẢO AN TOÀN THÔNG TIN TRONG LƯỚI TÍNH TOÁN

Do đặc điểm hỗn tạp và không đồng nhất của các tổ chức và tài nguyên trong lưới, vấn đề an toàn thông tin trong lưới là một trong những vấn đề được quan tâm hàng đầu. Có những vấn đề an toàn thông tin mới chưa từng gặp trong các công nghệ an toàn thông tin hiện tại cho hệ thống tính toán phân tán truyền thông. Ví dụ, các tính toán song song đòi hỏi nhiều tài nguyên tính toán dẫn tới nhu cầu phải thiết lập các mối quan hệ an toàn thông tin, không đơn giản chỉ là clien với server, mà giữa hàng trăm tiến trình thực hiện trong không gian tập hợp nhiều miền quản trị. Ngoài ra phải có các chính sách an toàn thông tin liên miền cho lưới, các công nghệ điều khiển truy nhập các miền khác nhau cũng phải được hỗ trợ.

Các ứng dụng và hệ thống lưới có thể đòi hỏi bất cứ chức năng nào trong các chức năng cơ bản của an toàn thông tin như là chứng thực, điều khiển truy nhập và toàn vẹn. Khi phát triển kiến trúc lưới, cũng cần phải lựa chọn giải pháp để đáp ứng được đòi hỏi của các đặc tính rất riêng của lưới.

- Đăng nhập một lần:

Khi bắt đầu một tính toán đòi hỏi sử dụng tài nguyên, cho thuê tài nguyên hay truyền thông nội bộ, người dùng có thể được chứng thực và sẽ không phải chứng thực trong tính toán tiếp theo.

- Giấy ủy nhiệm người dùng

Các mật khẩu, khóa bí mật phải được bảo vệ bằng các chính sách như mã hóa, hệ thống file bảo mật, phân quyền.

- Tích hợp các giải pháp an toàn thông tin địa phương

Các giải pháp liên miền phải tích hợp với các giải pháp an toàn thông tin địa phương để đảm bảo độc của các thành viên lưới.

- Hạ tầng giấy ủy nhiệm, chứng chỉ số thống nhất:

Truy nhập liên minh đòi hỏi phải có một quy ước thống nhất để biểu diễn định danh của các thực thể lưới như người dùng, tài nguyên…Vì thế, cần có một chuẩn để mã hóa các chứng chỉ số cho mục đích an toàn thông tin. Hiện tại, X509 là chuẩn cho các chứng chỉ số phổ biến trong môi trường lưới.

- Hỗ trợ an toàn nhóm truyền thông:

Một tính toán có thể đòi hỏi một số các tiến trình, cùng cộng tác các hoạt động của chúng với nhau như là một nhóm. Tổ chức các nhóm tiến trình sẽ thay đổi trong vòng đời của một tính toán. Vì thế cần cung cấp an toàn truyền thông nhóm động. Không có giải pháp nào hiện tại hỗ trợ tính năng này, thậm chí là thư viện lập trình GSS_API còn không cung cấp an toàn truyền thông nhóm.

- Độc lập công nghệ:

Các chính sách không phục vụ cho một công nghệ phát triển ứng dụng cụ thể nào. Hơn nữa, có thể cài đặt chính sách trong một phạm vi các công nghệ an toàn thông tin, dựa trên cả kỹ thuật mã hóa công khai và phân phối khóa công khai.

1.5.1. Cơ chế bảo đảm ATTT trong tính toán lưới

Các thành phần tham gia lưới lại chịu tác động của chính sách cục bộ trong phạm vi của mỗi thực thể tham gia lưới. Để giải quyết khó khăn này, cơ chế bảo đảm an toàn thông tin lưới cho phép tổ chức ảo dùng chung một phần chính sách với các tổ chức thực. Giải pháp tải chồng các chính sách như trên bắt buộc bảo đảm an toàn thông tin lưới phải đảm bảo các chức năng như: hỗ trợ nhiều cơ chế bảo mật khác nhau, khởi tạo động các dịch vụ, thiết lập động các miền chứng thực tin tưởng.

1.5.2. Các chính sách bảo đảm ATTT trong tính toán lưới

Sau đây là các chính sách bảo đảm an toàn thông tin:

- Môi trường lưới bảo đảm an toàn thông tin đa miền: tập trung điều khiển các tương tác liên miền, ánh xạ hoạt động liên miền với các chính sách bảo đảm an toàn thông tin địa phương.

- Hoạt động lưới hạn chế trong đơn miền quản trị: các hoạt động đa miền phải tuân theo chính sách bảo đảm an toàn thông tin địa phương trên miền quản trị đơn.

- Các chủ thể toàn cục và cục bộ đều tồn tại: tại mỗi miền quản trị đơn đều tồn tại hai chủ thể trên.

- Chứng thực đa phương: hoạt động giữa các thực thể trong các miền tin tưởng khác nhau đòi hỏi phải có chứng thực đa phương.

- Mỗi đối tượng toàn cục được ánh xạ vào đối tượng cục bộ đó được coi như chúng đã qua chứng thực địa phương trên đối tượng cục bộ đó.

- Tất cả các quyết định điều khiển được đưa ra đều là cục bộ hay dựa trên cơ sở của đối

tượng cục bộ.

- Có thể dùng chung tập giấy chứng nhận với các chương trình thay mặt cho cùng một tiến trình, chạy trên cùng một chủ thể trong cùng một miền tin tưởng.

1.5.3. Cơ sở Hạ tầng an ninh trong lưới tính toán

GSI là cơ chế cho phép xác thực và truyền thông an toàn trên mạng lưới. Nó cung cấp một số dịch vụ như: khả năng xác thực lẫn nhau, cơ chế đăng nhập một lần, cơ chế ủy quyền. GSI dựa trên các công nghệ mã khoá công khai (Public Key Infrastructure), Chứng thực X.509 (Certificate), nghi thức truyền thông bảo mật (Secure Socket Layer).

Những chuẩn công nghiệp về bảo đảm an toàn thông tin trên được thêm vào cơ chế đăng nhập một lần (SSO) và uỷ quyền (Proxy) tạo nên nền tảng bảo đảm an toàn thông tin vững chắc của mạng lưới. Sau đây là một số đặc điểm của GSI và các cài đặt ứng dụng của nó.

Cơ sở hạ tầng khóa công khai (PKI):

Chứng thực trong GSI là thao tác cung cấp cho mỗi thực thể một tên định danh duy nhất bằng cách đưa ra khái niệm giấy ủy quyền lưới, nó là một cặp giấy chứng nhận và khóa mã hóa (khóa bí mật). Trong môi trường PKI, mỗi thực thể phải trao quyền sở hữu khóa bí mật của mình để bảo đảm sự toàn vẹn của hệ thống.

Bảo mật mức thông điệp và mức giao vận:

GSI cho phép thực hiện bảo mật ở mức giao vận và mức thông điệp. Nếu chúng ta sử dụng bảo mật mức giao vận, toàn bộ truyền thông được mã hóa. Nếu sử dụng bảo mật mức thông điệp thì chỉ nội dung của thông điệp SOAP được mã hóa.

Hình 1.4. Bảo mật mức giao vận

Hình 1.5. Bảo mật mức thông điệp

Cả hai mức bảo mật này đều dựa trên khóa công khai, và do đó có thể đảm bảo tính toàn vẹn, riêng tư và khả năng chứng thực. Thường thì hội thoại an toàn phải đảm bảo tối thiểu khả năng chứng thực. Toàn vẹn thường rất cần thiết, nhưng có thể bỏ qua. Mã hóa có thể được kích hoạt để đảm bảo tính riêng tư.

Giấy ủy nhiệm

Trong môi trường lưới, người sử dụng cần được chứng thực nhiều lần trong khoảng thời gian tương đối ngắn. GSI giải quyết vấn đề này với khái niệm giấy ủy nhiệm. Mỗi giấy ủy nhiệm sẽ hoạt động thay mặt người dùng trong một khoảng thời gian ủy quyền ngắn hạn. Giấy ủy nhiệm có giấy chứng nhận và khóa bí mật riêng của nó, được tạo ra bằng cách kí lên giấy chứng nhận dài hạn của người dùng.

Giấy ủy nhiệm theo một cách khác, là một sự liên kết ngắn hạn giữa tên định danh của người dùng với một khóa bí mật khác. Chứng chỉ số thường được lưu trữ sử dụng mã hóa trong hệ thống file địa phương, thường được bảo vệ bởi quyền truy cập file trong hệ thống, có thể được sử dụng nhiều lần mà không cần sự bất tiện nào. Còn giấy ủy nhiệm dễ bị tổn thương, nó có thời gian sống ngắn hạn hơn nhiều so với các chứng chỉ số dài hạn của người dùng, thông thường là vài giờ.

Sự ủy quyền

Các ứng dụng của người dùng có thể thay mặt họ trong môi trường lưới. GSI cho phép người dùng ủy quyền giấy ủy nhiệm của mình để giao dịch các máy từ xa.

Sự ủy quyền cũng tương tự như việc tạo ra các giấy ủy nhiệm, mỗi tập chứng chỉ số dài hạn sẽ được dùng để tạo ra tập các giấy ủy nhiệm mới, có thời gian sống ngắn hơn. Sự khác nhau là việc tạo ra các giấy ủy nhiệm xảy ra trong các phiên kết nối đòi hỏi chứng thực GSI, khi các tiến trình từ xa đòi hỏi giấy ủy nhiệm của người dùng cho chứng thực. Một điều đáng chú ý nữa là sự ủy quyền có thể là một chuỗi, một người có thể ủy quyền cho một máy A, sau đó tiến trình sử dụng trên máy A có thể ủy quyền cho máy B và cứ tiếp tục như vậy.

Chứng thực

GSI hỗ trợ cơ chế cho phép chuyển các tên định danh GSI của người dùng vào trong các định danh địa phương (tài khoản của một người dùng Unix cục bộ). Việc chứng thực các định danh GSI sẽ chuyển về chứng thực các định danh địa phương, cùng với việc đó, các chính sách đưa ra cũng nằm trong phạm vi cục bộ như: quyền truy nhập file, dung lượng đĩa, tốc độ CPU.

Xác thực và quyền hạn

Để có thể đảm bảo rằng đối tác trong phiên truyền thông trong mạng lưới là đối tác tin cậy, ta có thể sử dụng chức năng xác thực của GSI. Sau khi ta đã được xác thực có tài nguyên mạng lưới yêu cầu phải xác định các quyền truy cập. Khi đó ta có thể sử dụng chức năng kiểm tra

quyền của GSI. Sau đây ta sẽ mô tả các bước để xác thực, kiểm tra quyền của một máy A (người dùng trên máy A) bởi một máy B trong mạng lưới. Hầu hết các bước để xác thực, trừ bước cuối cùng để kiểm tra quyền hạn.

- Bước 1: người dùng trên máy A hay một ứng dụng trên máy A gửi chứng nhận của nó tới máy B.

- Bước 2: máy B sẽ sử dụng khóa công khai của A để trích ra tiêu đề từ chứng nhận được gửi ở bước 1.

- Bước 3: Máy B sinh ra một số ngẫu nhiên và gửi lại cho máy A.

- Bước 4: Máy A nhận được số ngẫu nhiên đó rồi sử dụng khóa riêng của mình để mã hóa số nhận được và gửi kết quả lại cho B.

- Bước 5: Máy B giải mã nhận được kết quả là một số, rồi kiểm tra số này với số đã sinh ra ở bước 3. Nếu 2 số này là bằng nhau thì máy B đã biết rằng chứng nhận mà nó nhận được đúng là của người dùng trên máy A. Vì chỉ có anh ta mới có khóa bí mật của mình.

- Bước 6: Chứng nhận đã được xác thực tại máy B, và tiêu đề của xác thực được map tương ứng với một tên người dùng trên máy B. Tiêu đề ở dạng tên phân biệt. (distinguished name). Tiêu đề này được sử dụng để xác định định danh của người dùng trong môi trường mạng lưới. Người dùng được xác định từ tiêu đề sẽ được máy B kiểm tra quyền như một người dùng trên máy B.

Chương 2. MỘT SỐ BÀI TOÁN VỀ AN TOÀN THÔNG TIN TRONG TÍNH TOÁN LƯỚI.

Bảo vệ thông tin trong lưới tính toán cũng như trong các hệ thống thông tin nói chung gồm 4 nhiệm vụ chính:

Bảo mật, bảo toàn, Xác thực, Sẵn sàng.

Nhưng trong đồ án này em xin tập trung vào hai bài toán: Bảo mật và Xác thực

2.1. BÀI TOÁN XÁC THỰC THỰC THỂ SỬ DỤNG LƯỚI TÍNH TOÁN

- **Xác thực** là một trong những vấn đề nóng bỏng. Xác thực là xác minh, kiểm tra một thông tin hay một thực thể nào đó để công nhận hoặc bác bỏ tính hợp lệ của thông tin hay thực thể đó. Đây là yêu cầu rất quan trọng trong các giao tiếp cần có sự tin cậy.

Xác thực bao gồm 2 việc chính:

+ Xác thực tính hợp lệ của các thực thể tham gia giao tiếp.

+ Xác thực nguồn gốc của thông tin được trao đổi.

- **Xác thực điện tử** là việc xác minh từ xa bằng các phương tiện điện tử sự tồn tại chính xác và hợp lệ danh tính của chủ thể nào đó, cũng như thông tin nào đó mà không cần biết nội dung cụ thể của thông tin và chủ thể đó.

Mục đích của việc xác thực điện tử: chống giả mạo, chống chối bỏ, tính xác thực của thông tin và mục đích cuối cùng là hoàn thiện các giải pháp an toàn thông tin.

Cơ sở để xây dựng các giải pháp cho xác thực điện tử là các hệ mật mã.

Giải pháp xác thực thực thể sử dụng lưới

+ Xác thực chữ ký điện tử.

+ Xác thực dấu hiệu bản quyền.

Nhưng trong chương này em xin tập trung vào xác thực chữ ký.

2.1.1 Khái niệm chữ ký số

Một sơ đồ chữ ký l tập (P, A, K, S, V) thoả mãn các điều kiện dưới đây:

+ P là tập hữu hạn các bức điện (thông điệp)

+ A là tập hữu hạn các chữ kí

+ K là tập hữu hạn các khóa, K(k',k")

Trong đó: Khóa k': khóa bí mật (khóa ký)

Khóa k": khóa công khai (khóa kiểm thử)

$Sig_{k'}$ là thuật toán ký $P \rightarrow A$

$$x \in P \rightarrow y = Sig_{k'}(x)$$

$Ver_{k''}$ là thuật toán kiểm thử: (P, A) $\rightarrow$ (Đúng, sai)

$Ver_{k''}(x, y)$ = | Đúng Nếu $y = Sig_{k'}(x)$

| Sai Nếu $y \neq Sig_{k'}(x)$

2.1.1.1. Chữ ký RSA

+ ***Tạo khóa:***

Sơ đồ chữ ký cho bởi bộ năm (P, A, K, S, V)

Cho n=p.q; với mỗi p, q là các số nguyên tố lớn khác nhau $\phi(n) = (p - 1)(q - 1)$.

Cho $P = A = Z_n$ và định nghĩa:

K là tập các khóa, K=(k',k''); với k'=a: khoá bí mật

k''=(n, b) khóa công khai

Trong đó: a, b $\in Z_n^*$, thỏa mãn:

b<$\phi(n)$ và nguyên tố cùng nhau với $\phi(n)$.

a<$\phi(n)$ sao cho: $a.b \equiv 1 \bmod \phi(n)$.

Các giá trị n, b là công khai, các giá trị p, q, a là các giá trị bí mật.

+ ***Tạo chữ ký***:

Với mỗi K=(n, p, q, a, b) chữ ký trên x là:

$$y = Sig_{k'}(x) = x^a \bmod n$$

+ ***Kiểm tra chữ ký***:

$Ver_{k''}(x, y)$= true $\Leftrightarrow x \equiv y^b \bmod n$; x, y $\in Z_n$.

Giả sử A muốn gửi thông báo x, A sẽ tính chữ ký y bằng cách :

$y=sig_{k'}(x) = x^a \bmod n$ (a là khóa bí mật của A)

A gửi cặp (x, y) cho B. Nhận được thông báo x, chữ ký số y, B bắt đầu tiến hành kiểm tra đẳng thức:

$x = y^b \bmod (n)$ (b là khóa công khai A)

Nếu đúng, B công nhận y là chữ ký trên x của A. Ngược lại, B sẽ coi x không phải của A gửi cho mình .

Ví dụ

A dùng lược đồ chữ ký số RSA với n=15,(p=3, q=5);

$\phi(n) = 2.4 = 8$

Chọn b=3.

$\Rightarrow a = 3^{-1} \bmod 8 = 3$ vì $(3.3 \equiv 1 (\bmod 8))$

Khóa ký k'=3

Khóa công khai k''=(15, 3)

A ký trên thông báo x=2 với chữ ký:

$y = x^a \bmod n = 2^3 \bmod 15 = 8$

A gửi cặp (x, y) = (2, 8) cho B, B kiểm tra bằng cách sử dụng khóa công khai của A như sau:

$x = y^b \bmod n = 8^3 \bmod 15 = 2 = x$.

B chấp nhận y=8 là chữ ký tin cậy.

2.1.1.2. Chữ ký ElGamal

Chữ ký ElGamal được định nghĩa như sau:

+ ***Tạo khóa***:

Cho p là số nguyên tố sao cho bài toán logarit rời rạc trong Z_p là khó và giả sử $\alpha \in Z_p^*$, là phần tử nguyên thủy

Cho $P = Z_p^*$, $A = Z_p^* \times Z_{p-1}$ và định nghĩa

$K = \{(p, a, \alpha, \beta): \beta = \alpha^a \bmod p\}$.

Các giá trị p, α, β là công khai, a là bí mật.

+ ***Tạo chữ ký***

Với K = (p, a, α, β) và với số ngẫu nhiên k $\in Z_p^{*}{}_{-1}$

định nghĩa $sig_k(\gamma, \delta)$, trong đó:

$$\gamma = \alpha^k \text{ mod} p \text{ và } \delta = (x - a\gamma)\, k^{-1} \text{ mod } (p - 1).$$

+ ***Kiểm tra chữ ký số***

Với x, $\gamma \in Z_p^{*}$ và $\delta \in Z_{p-1}$, ta định nghĩa :

$$\text{Ver } (x, \gamma, \delta) = \text{True} \Leftrightarrow \beta^{\gamma}.\, \gamma^{\delta} \equiv \alpha^{x} \text{ mod } p.$$

Chứng minh:

Nếu chữ ký được thiết lập đúng thì hàm kiểm tra sẽ thành công vì:

$$\Rightarrow \beta^{\gamma}\, \gamma^{\delta} \equiv \alpha^{a.\gamma}\, \alpha^{r.\delta} \text{ mod } p$$

$$\equiv \alpha^{x} \text{ mod } p \text{ (vì } a\gamma + r\delta \equiv x \text{ mod } (p - 1)).$$

A tính chữ ký bằng cách dùng cả giá trị bí mật a (là một phần của khóa) lẫn số ngẫu nhiên bí mật k (dùng để ký trên x). Việc kiểm tra có thể thực hiện duy nhất bằng thông tin công khai.

Ví dụ: Giả sử p=467, α = 2, a = 127

Khi đó: $\beta = \alpha^a \text{ mod } p = 2^{127} \text{mod } 467 = 132$

Giả sử A có thông báo x=100 và A chọn ngẫu nhiên k=213 vì (213,466)=1

và $213^{-1} \text{ mod } 466 = 431$, A ký trên x như sau:

$$\gamma = \alpha^k \text{ mod } p = 2^{213} \text{mod } 467 = 29$$

Và $\delta = (x - a\gamma)k^{-1} \text{ mod } (p - 1) = (100 - 127.\, 29).431 \text{ mod } 466 = 51.$

Chữ ký của A trên x= 100 là (29,51).

Bất kỳ người nào đó cũng có thể kiểm tra chữ ký bằng cách tính:

$$132^{29}\, .\, 29^{51} \equiv 189 \text{ mod } 467$$

$$2^{100} \equiv 189 \text{ mod } 467$$

Do đó, chữ ký là tin cậy.

2.1.2. Sử dụng chữ ký số trong xác thực thực thể dùng lưới tính toán

Chữ kí số là đoạn dữ liệu gắn liền với văn bản gốc để chứng thực tác giả của văn bản và giúp người nhận kiểm tra xem chữ ký có bị giả mạo hay không bằng cách sử dụng công nghệ khóa công khai PKI (Public Key Infrastructure). Trong đó mỗi người tham gia ký cần một cặp khóa bao gồm một khóa công khai và một khóa bí mật. Khóa bí mật dùng để tạo chữ ký số, khóa công khai dùng để thẩm định, xác thực chữ ký số.

2.2. BÀI TOÁN BẢO MẬT TRONG LƯỚI TÍNH TOÁN

Các thông tin truyền trên lưới cần phải được bảo mật bằng mã hóa, do đó trong mục này em sẽ trình bày về Hệ mã hóa.

2.2.1. Khái niệm mã hóa

Hệ mã hóa là hệ bao gồm 5 thành phần (P, C, K, E, D) thỏa mãn các tính chất sau:

P (Plaitext): là tập hợp hữu hạn các bản rõ có thể.

C (Ciphertext): Là tập hữu hạn các bản mã có thể.

K (Key): Là tập hợp các bản khoá có thể.

E (Encrytion): Là tập hợp các quy tắc mã hoá có thể.

D (Decrytion): Là tập hợp các quy tắc giải mã có thể.

Chúng ta đã biết một thông báo thường được xem là bản rõ. Người gửi sẽ làm nhiệm vụ mã hoá bản rõ, kết quả thu được gọi là bản mã. Bản mã được gửi đi trên đường truyền tới người nhận. Người nhận giải mã để tìm hiểu nội dung bản rõ. Dễ dàng thấy được công việc trên khi định nghĩa hàm lập mã và hàm giải mã:

$E_k(P) = C$ và $D_k(C) = P$

2.2.1.1. Hệ mã hóa khóa đối xứng

Mã hóa khóa đối xứng là hệ mã hóa mà biết chìa lập mã "dễ" tính ra chìa giải mã và ngược lại. Trong một số hệ mã hóa đối xứng thì hai khóa này là trùng nhau.

Mặc dù các phương pháp mã hóa đối xứng thường có tốc độ cao và dễ cài đặt, nhưng chúng lại có nhiều yếu điểm. Một nhược điểm chính đó là vì cả người gửi và người nhận đều sử dụng cùng một khóa mã hóa do đó cần phải có sự trao đổi thông tin thống nhất khóa thông qua một kênh mật. Đây là một vấn đề khó trong an toàn và bảo mật.

Các hệ thống bảo vệ thông tin ngày nay sử dụng các thuật toán mã hóa bất đối xứng để cùng với hệ mã hóa đối xứng bảo vệ thông tin.

Sau đây là một số hệ mật mã cổ điển:

1/. Mã dịch chuyển:

Định nghĩa: Mã dịch chuyển: (P, C, K, E, D)

$P = C = K = Z_{26}$ với $k \in K$, định nghĩa $e_k(x) = (x + k) \bmod 26$ $d_k(y) = (y – k) \bmod 26$

$(x, y \in Z_{26})$

Ví dụ: Dùng khoá k = 9 để mã hoá dòng thư"toinaydichoi" dòng thư đó tương ứng với dòng số:

t	o	i	n	a	y	d	i	c	h	o	i
19	14	8	12	0	24	3	8	2	7	14	8

qua phép mã hoá e_9 sẽ được:

2	23	17	22	9	7	12	17	11	16	23	17

c	x	r	w	j	h	m	r	l	q	x	r

bản mã sẽ là:

"qnwcxrcqdkjh"

Nhận được bản mã đó, dùng d_9 để nhận được bản rõ.

2/. Mã thay thế:

Định nghĩa Mã thay thế: (P, C, K, E, D)

$P = C = Z_{26}$, $K = S(Z_{26})$ Với mỗi $\pi \in K$, tức là một hoán vị trên Z_{26}, ta xác định

$$e_\pi(x) = \pi(x)$$

$$d_\pi(y) = \pi^{-1}(y)$$

với $x, y \in Z_{26}$, π^{-1} là nghịch đảo của л

Ví dụ: π được cho bởi (ở đây ta viết chữ cái thay cho các con số thuộc Z_{26}):

a	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*j*	*k*	*l*	*m*	*n*
x	*n*	*y*	*a*	*h*	*p*	*o*	*g*	*z*	*q*	*w*	*b*	*t*	*s*

o	*p*	*q*	*r*	*s*	*t*	*u*	*v*	*w*	*x*	*y*	*z*
f	*l*	*r*	*c*	*v*	*m*	*u*	*e*	*k*	*j*	*d*	*i*

bản rõ:

"toinaydichoi"

sẽ được mã hoá thành bản mã (với khoá π):

"mfzsxdazygfz"

Dễ xác định được π^{-1}, và do đó từ bản mã ta tìm được bản rõ.

Mã thay thế có tập hợp khoá khá lớn - bằng số các hoán vị trên bảng chữ cái, tức số các hoán vị trên Z_{26}, hay là 26! > 4.10^{26}. Việc duyệt toàn bộ các hoán vị để thám mã là rất khó, ngay cả đối với máy tính. Tuy nhiên, bằng phương pháp thống kê, ta có thể dễ dàng thám được các bản mã loại này, và do đó mã thay thế cũng không thể được xem là an toàn.

*3/. <u>Mã Anffine</u>***:**

Định nghĩa: Mã Anffine: (P, C, K, E, D)

$$P = C = Z_{26}, K = \{ (a, b) \epsilon Z_{26} \text{ x } Z_{26} : (a, 26) = 1 \}$$

với mỗi k = (a, b) ϵ K ta định nghĩa:

$$e_k(x) = ax + b \bmod 26$$

$$d_k(y) = a^{-1}(y - b) \bmod 26 , \quad \text{trong đó } x, y \epsilon Z_{26}.$$

Ví dụ: Lấy k = (5, 6).

Bản rõ:

"toinaydichoi"

	t	o	i	n	a	y	d	i	c	h	o	i
x	19	14	8	13	0	14	3	8	2	7	14	8

y=5x + 6 mod 26

y	23	24	20	19	6	24	21	20	16	15	24	20
	x	y	u	t	g	y	v	u	q	p	y	u

Bản mã:

"xyutgyvuqpyu"

Thuật toán giải mã trong trường hợp này có dạng:

$$d_k(y) = 21(y - 6) \bmod 26$$

Với mã Apphin, số các khoá có thể có bằng (số các số ≤ 26 và nguyên tố với 26) × 26, tức là 12 × 26 = 312. Việc thử tất cả các khoá để thám mã trong trường hợp này tuy khá mất thì giờ nếu tính bằng tay, nhưng không khó khăn gì nếu dùng máy tính. Do vậy, mã Apphin cũng không phải là mã an toàn.

4/. <u>Mã hoán vị</u>:

Định nghĩa Mã hoán vị: (P, C, K, E, D)

Cho m là số nguyên dương.

$P = C = Z_{26}$, $K = S_m$ với mỗi $k = \pi \in S_m$, ta có:

$$e_k(x_1, x_2, \ldots, x_m) = (x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(m)})$$

$$d_k(y_1, y_2, \ldots, y_m) = (y_{\pi^{-1}(1)}, y_{\pi^{-1}(2)}, \ldots, y_{\pi^{-1}(m)})$$

trong đó π^{-1} là hoán vị nghịch đảo của π

Ví dụ: Giả sử m = 6, và khoá k được cho bởi phép hoán vị π

1	2	3	4	5	6
3	5	1	6	4	2

Khi đó phép hoán vị nghịch đảo π^{-1} là:

1	2	3	4	5	6

3	6	1	5	2	4

Bản rõ:

“toinaydichoi”

	t	o	i	n	a	y	d	i	c	h	o	i
vt	1	2	3	4	5	6	1	2	3	4	5	6
π	1->3	2->5	3->1	4->6	5->4	6->2	1->3	2->5	3->1	4->6	5->4	6->2
vt	3	5	1	6	4	2	3	5	1	6	4	2
	i	a	t	y	n	o	c	o	d	i	h	i

Bản mã:

“iatynocodihi”

Dùng hoán vị nghịch đảo, từ bản mật mã ta lại thu được bản rõ.

2.2.1.2. Hệ mã hóa khóa phi đối xứng

Mã hóa khóa phi đối xứng là hệ mã hóa mà biết chìa này “khó” tính được chìa kia. Các khóa này được xây dựng bằng hàm một chiều có cửa sập.

Trong hai khóa đó, một khóa được chọn làm khóa bí mật và khóa còn lại được chọn làm khóa công khai. Khóa bí mật chỉ có một người là chủ nhân của nó nắm giữ. Khóa công khai được công bố rộng rãi cho bất cứ ai muốn trao đổi thông tin mật với người sở hữu khóa. Khóa công khai được sử dụng để mã hóa thông tin và khóa bí mật được sử dụng để giải mã.

Quá trình giao tiếp giữa 2 đối tượng A và B được mô tả như sau:

B sinh ra cặp khóa bí mật và công khai, khóa bí mật được cất giữ một cách an toàn và được bảo vệ bằng một mật mã còn khóa công khai được cung cấp rộng rãi. A có thể sử dụng khóa công khai (được phát hành bởi B) để mã hóa thông tin và gửi cho B. Lúc này, chỉ duy nhất B, người sở hữu khóa bí mật, có thể giải mã thông tin bằng khóa bí mật.

Ngoài ra, mã hóa phi đối xứng còn được dùng trong các cơ chế xác thực. Tuy nhiên, một nhược điểm lớn của hệ mã hóa công khai này là quá trình giải mã cũng như mã hóa mất nhiều thời gian.

2.2.2. Sử dụng hệ mã hóa trong bảo mật thông tin trên lưới tính toán

Ta biết rằng tin truyền trên mạng rất dễ bị lấy cắp, thay đổi…. Để đảm bảo việc truyền tin an toàn người ta thường mã hoá thông tin trước khi truyền đi. Việc mã hoá thường theo quy tắc nhất định gọi là hệ mật mã. Các hệ thống thông tin ngày nay thường sử dụng hệ mã hóa phi đối xứng vì độ an toàn cao hơn.

Sau đây là một số hệ mã hóa phi đối xứng thường dùng.

2.2.2.1. Hệ mã hoá RSA

+ Sinh khoá:

Cho n=p*q với p, q là số nguyên tố lớn. Đặt $P = C = Z_n$

Chọn khoá công khai b: Chọn $b<\phi(n)$, nguyên tố cùng nhau với $\phi(n)$, $\phi(n) = (p-1)(q-1)$

Chọn khoá bí mật a: Chọn a là nghịch đảo của b theo modulo $\phi(n)$: $a*b \equiv 1(\text{mod } \phi(n))$

K=(k', k'') trong đó: k'= (n, b) khóa công khai

k''= a khóa bí mật

Với mỗi khoá K=(n, a, b), mỗi $x \in P$, $y \in C$ định nghĩa

+ Hàm lập mã : $y = e_{k'}(x) = x^b \bmod n$

+ Hàm giải mã: $d_{k''}(y) = y^a \bmod n$

2.2.2.2 Hệ mã hoá ElGamal.

Hệ mật mã ElGamal được T.ElGamal đề xuất năm 1985, dựa vào độ phức tạp của bài toán tính lôgarit rời rạc, và sau đó đã nhanh chóng được sử dụng rộng rãi không những trong vấn đề bảo mật truyền tin mà còn trong các vấn đề xác nhận và chữ ký điện tử.

Bài toán logarithm rời rạc trong Z_p là đối tượng trong nhiều công trình nghiên cứu và được xem là bài toán khó nếu p được chọn cẩn thận. Cụ thể là không có một thuật toán thời gian đa thức nào cho bài toán logarithm rời rạc. Để gây khó khăn cho các phương pháp tấn công đã

biết, p phải có ít nhất 150 chữ số và (p – 1) phải có ít nhất một thừa số nguyên tố lớn. Hệ mật Elgamal là một hệ mật không tất định vì bản mã phụ thuộc vào cả bản rõ x lẫn giá trị ngẫu nhiên k do G chọn. Bởi vậy sẽ có nhiều bản mã được mã từ cùng một bản rõ.

Bài toán logarithm rời rạc trong Z_p:

Đặc trưng của bài toán: I = (p, α, β) trong đó p là số nguyên tố, $\alpha \in Z_p$ là phần tử nguyên thuỷ (hay phần tử sinh), $\beta \in Z_p^*$.

Bài toán: Tìm x sao cho $\alpha^x \equiv \beta \pmod p$.

Định nghĩa mã khóa công khai Elgamal trong Z_p^*:

+ Sinh khoá:

Cho p là số nguyên tố sao cho bài toán logarithm rời rạc trong Z_p là khó giải.

Cho $\alpha \in Z_p^*$ là phần tử nguyên thuỷ.

Giả sử $P = Z_p^*$, $C = Z_p^* \times Z_p^*$.

Ta định nghĩa: K = {K(k',k'');k'=(p, α, a, β);k''=a: $\beta \equiv \alpha^a \pmod p$}

Các giá trị p, α, β được công khai, còn a giữ kín.($0 \le a \le p-2$)

Với K =(p, α, a, β) và một số ngẫu nhiên bí mật $k \in Z_{p-1}$, ta xác định:

+ Lập mã:

$$E_{k'}(x, k) = (y_1, y_2)$$

Trong đó:

$$y_1 = \alpha^k \bmod p$$

$$y_2 = x.\ \beta^k \bmod p$$

Với $y_1, y_2 \in Z_p^*$:

+ Giải mã:

$$d_{k''}(y_1, y_2) = y_2(y_1^a)^{-1} \bmod p$$

Ví dụ:

Chọn p = 7

$\alpha \in Z_p^*$ là phần tử nguyên thuỷ nên $\alpha = 3$

Chọn a sao cho $0 \le a \le p - 2$ nên chọn a = 2

Khi đó: $\beta = \alpha^a \bmod p = 3^2 \bmod 7 = 2$

Chọn x sao cho: $3^x \equiv \beta \pmod p \Rightarrow x=3$

Chọn một số ngẫu nhiên bí mật $k \in Z_{p-1}$, chọn k =3

Giả sử A muốn gửi thông báo x = 3 cho B, A phải tính:

$E_{k'}(x, k) = (y_1, y_2)$

trong đó: $y_1 = \alpha^k \bmod p = 3^3 \bmod 7 = 6$

$y_2 = x.\ \beta^k \bmod p = 3*2^3 \bmod 7 = 3$

Khi B thu được bản mã $(y_1, y_2) = (6, 3)$, anh ta sẽ tính:

$$d_{k''}(y_1, y_2) = y_2(y_1^a)^{-1} \bmod p = 3*(6^2)^{-1} \bmod 7 = 3=x$$

Đó chính là bàn rõ mà G đã mã hoá.

Chương 3. THỬ NGHIỆM CHƯƠNG TRÌNH KÝ SỐ TRONG TTL

Trong chương 3, đồ án thử nghiệm thử viết chương trình ký số RSA.

3.1. CẤU HÌNH HỆ THỐNG

+ **Phần mềm:** Tubo C++ 3.0

-Ưu điểm: miễn phí, không cần cài đặt, biên dịch và chạy chương trình nhanh, môi trường tích hợp thuận tiện.

-Nhược điểm: không thể biên dịch chương trình chạy trên window, không hỗ trợ các công nghệ mới như nhắc nhở người dùng các từ khoá, hàm và kiểu dữ liệu, thao tác soạn thảo của Tubo C++3.0 cũng không tiện lợi vì đòi hỏi sử dụng các tổ hợp phím khá phức tạp…

+ **Phần cứng**: chiếm dung lượng nhỏ (khoảng 4,3 MB), có thể chạy trên mọi thế hệ của máy tính có hệ điều hành DOS.

3.2. CÁC THÀNH PHẦN TRONG CHƯƠNG TRÌNH

Gồm 3 phần:

- Sinh khoá:

Input: hai số nguyên tố p, q, tính n=p.q, $\phi(n)= (p-1)(q-1)$

Output: Cặp khóa (bí mật, công khai) = (a, b)

- Ký số:

Input: khóa bí mật, bản rõ x.

Out put: Chữ ký số.

- Kiểm tra chữ ký:

Input: bản rõ x, chữ ký số, khóa công khai và n.

Output: Xác thực chữ ký đúng hoặc sai.

3.3. CHƯƠNG TRÌNH

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
#include <stdlib.h>
#include<string.h>
//=========================================
long int kha_nghich(long int b, long int n);
long exp_mod(long x, long b, long n);
int kiemtra_ngto(long pq);
long  USCLN(long  n,long m);
long Ktra_ngto_cungnhau(long b,long phi_N);
long Kitep(int Ki);
long Doctep(long n);
void Ky_RSA();
//===========Tinh Kha nghich ================
long int kha_nghich(long int b, long int n)
   {
        long int n0, b0;
        long int t, t0, temp, q, r;
        n0=n;   b0=b;   t0=0;    t=1;
        q=floor(n0/b0);
        r=n0-q*b0;
        while(r>0){
                        temp=t0-q*t;
                        if (temp < 0)
```

```
                    temp = n- ((-temp) % n);
                else
                    temp = temp % n;
                t0=t;
                t=temp;
                n0=b0;
                b0=r;
                q=floor(n0/b0);
                r=n0-q*b0;
            }
        if(b0!=1)
        {
            printf("Khong co a"); return 0;
        }
        else return(t%n);
    }
//=========== Tinh Mod ============
long exp_mod(long x, long b, long n)
{
        long a = 1l, s = x;
  while (b != 0) {
    if (b & 1l) a = (a * s) % n;
          b >>= 1;
        if (b != 0) s = (s * s) % n;
}
        if (a < 0) a += n;
 return a;
```

```
}
//============================================================
int kiemtra_ngto(long pq)
{
   for(long i=2;i<=(long)sqrt(pq);i++)
    if(pq%i==0)
       {
       printf("\n\n Khong phai so nguyen to!\n\nMoi ban nhap lai!");
       return 0;
       }
     return 1;
}
//============================================================
long  USCLN(long  n,long m)
{
    while(m!=0&&n!=0)
      if(n>m) n=n-m;
      else m=m-n;
      if(n==0) return m;
      else return n;
}
//============================================================
long Ktra_ngto_cungnhau(long b,long phi_N)
{
     if(USCLN(b,phi_N)!=1)
      {
          printf("\n\nb khong phai la nguyen to cung nhau voi phi_N\n\n moi chon lai b!");
```

```
            return 0;
       }
   else return 1;
}
//============================================================
long Kitep(int Ki)
{
    FILE *f;
   char *tentep;
   long n;
      mt:printf("\n\nNhap vao ten tep can Ki:");fflush(stdin);gets(tentep);
      f=fopen(tentep,"a+t");
   if(f==NULL)
    {
      printf("\n\nTep %s khong ton tai! Moi nhap lai!",tentep);
      getch();
    goto mt;
     }
            fseek(f,0,SEEK_END);
            n=ftell(f);
            fseek(f,n,SEEK_SET);
            fprintf(f,"%d",Ki);
            fclose(f);
    return n;
}
//============================================================
long Doctep(long n)
```

```
{
    FILE *f;
   char *tentep;
         mt:printf("\n\nNhap vao ten tep can mo:");fflush(stdin);gets(tentep);
         f=fopen(tentep,"a+t");
   if(f==NULL)
    {
         printf("\n\nTep %s khong ton tai! Moi nhap lai!",tentep);
   goto mt;
    }
      long ki;
      fseek(f,n,SEEK_SET);
      fscanf(f,"%ld",&ki);
      fclose(f);
   return ki;
}
//============================================================


void Ky_RSA()
{
  clrscr();
  long x,a,b,n,phi_N,p,q;
  long Kthuocvb;
  int Ki,Kiem_thu;
      printf("\n=====* CHU KY RSA *======");
      p:printf("\nNhap so nguyen to p=");scanf("%ld",&p);
  if(kiemtra_ngto(p)!=1)goto p;
```

```
    q:printf("\nNhap so nguyen to q=");scanf("%ld",&q);
  if(kiemtra_ngto(q)!=1)goto q;
            n=p*q;
            phi_N=(p-1)*(q-1);
    b:printf("\nMoi ban chon so b  (1<b<phi_N) sao cho gcd(b,phi_N)==1\n\n  b=");
    scanf("%ld",&b);
  if(Ktra_ngto_cungnhau(b,phi_N)!=1)goto b;
    a=kha_nghich(b,phi_N);
    printf("\n\n          LAP CHU KI ");
    printf("\nKhoa bi mat dung de tao chu ki la K1(a)=%ld",a);
    printf("\nNhap ban ro x=");scanf("%ld",&x);
            Ki=exp_mod(x,a,n);
    printf("\nVoi so x ta tao duoc ra chu Ki la :%d",Ki);
            Kthuocvb=Kitep(Ki);
    printf("\nVan ban da duoc ki!");
    printf("\n\n          KIEM THU CHU KI ");
    printf("\nKiem thu voi khoa cong khai la K2(b,n)=(%ld,%ld)",b,n);
            Kiem_thu=Doctep(Kthuocvb);
    printf("\nChu ki duoc lay tu tep la:%d",Kiem_thu);
    printf("\nKiem thu chu ki so ta duoc x=%d ",exp_mod(Kiem_thu,b,n));
  if(exp_mod(Kiem_thu,b,n)==x)
        printf("\n\n      CHU KI TREN LA DUNG!");
  else
        printf("\n\n      KHONG PHAI LA CHU KI!");
 getch();
}
//============================================================
```

```
void menu()
{
    int c;
    while(1)
     {
          clrscr();
          printf("\n\n=====* CHUONG TRINH CHU KY SO *=======");
          printf("\n\n[1].CHU KY RSA");
          printf("\n[2].Thoat khoi chuong trinh");
          printf("\n\n Moi ban chon:");scanf("%d",&c);




     switch(c)
          {
          case 2:
               return;
          case 1:
                    Ky_RSA();
                    break;
          }

  }
}
//=========================================
void main()
```

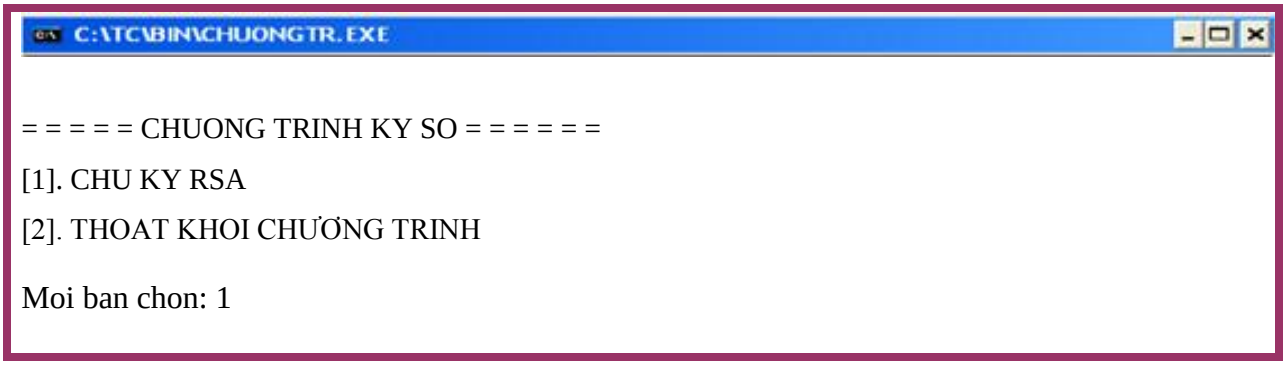
```
{clrscr();
 menu();
}
//================== Ket thuc ===============
```

3.4. HƯỚNG DẪN SỬ DỤNG CHƯƠNG TRÌNH

+Khởi động TC để vào chương trình.

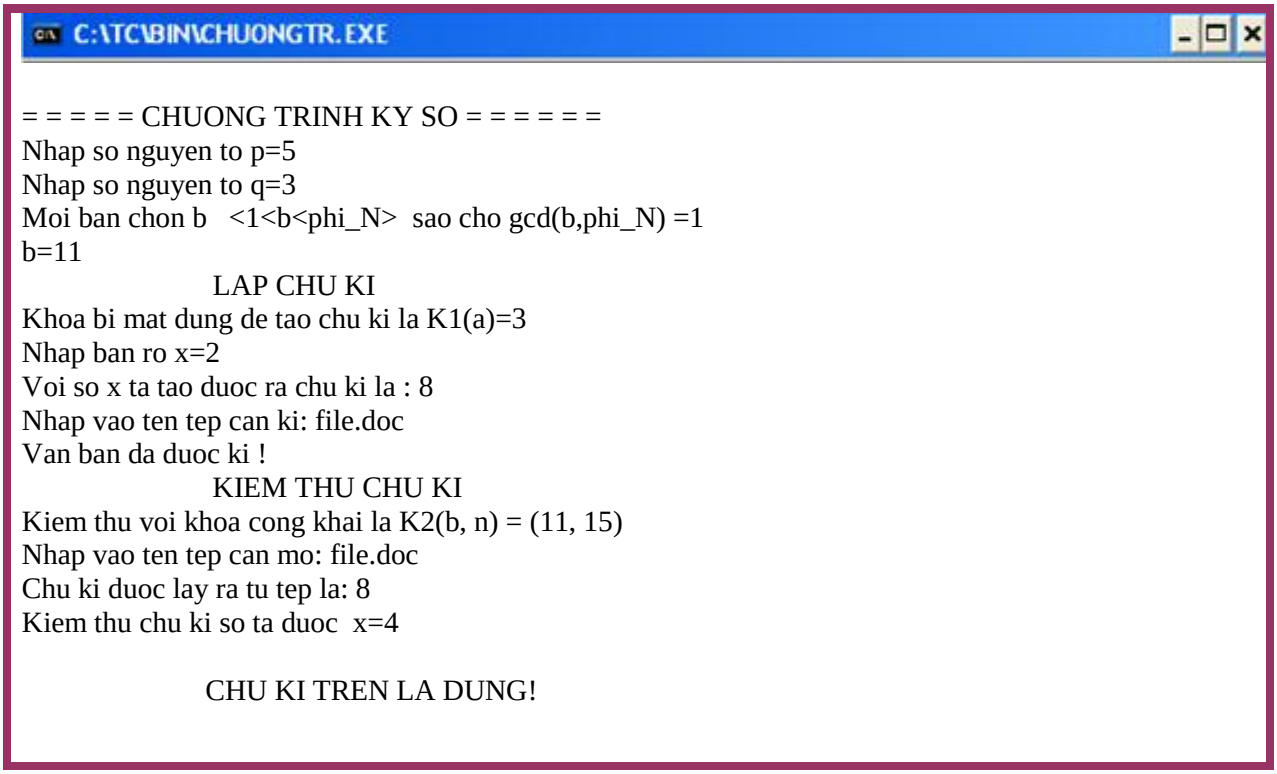
+Trước khi chạy chương trình nhấn phím F9 để kiểm tra lỗi.

+Nếu không báo lỗi nhấn tổ hợp phím Ctrl + F9 để chạy chương trình, xuất hiện giao diện như sau:



+ Nhấn phím 1 để vào chương trình, phím 2 để thoát khỏi chương trình.

+ Kết quả thử nghiệm chương trình:



+ Chú ý: Khi chương trình yêu cầu nhập tên tệp để ký:

- Nếu tệp cần ký và chương trình chạy được đặt trong cùng một thư mục thì ta chỉ việc nhập tên tệp văn bản cần ký.

- Ngược lại thì ta cần phải chỉ rõ đường dẫn đến tệp văn bản cần ký.

KẾT LUẬN

Đồ án T T gồm hai kết quả chính:

1/. Tìm hiểu và nghiên cứu qua tài liệu để hệ thống lại các vấn đề sau:

+ Tổng quan về tính toán lưới.

+ Các thành phần cơ bản của lưới tính toán.

+ Một số bài toán về an toàn thông tin trong tính toán lưới.

2/. Thử nghiệm chương trình

TÀI LIỆU THAM KHẢO

1/. Tiếng Việt:

[1] Phan Đình Diệu: *Lý thuyết mật mã và An toàn thông tin*, 2004.

[2] Trịnh Nhật Tiến: *Bài giảng môn An toàn dữ liệu*, 2005.

2/. Tiếng Anh:

[3] Ahmar Abbas. *Grid Computing: A Practical Guide to Technology and Applications*. Charles River Media, 2003.

[4] B.Jacob, M.Brown, K.Fukui, N.Trivedi. *Introduction to Grid Computing*. IBM Redbook, 2005.

[5] Borja Sotomayor. *Ther Globus Tookit 4 Programmer's Tutorial*. University of Chicago, 2005.

[6] Joshy Joseph, Craig Fellenstein. *Grid Computing*. IBM Press, 2003.

[7] Website: http:// www.toantin.org. Grid fundamental.

[8] Website: http://primes.utm.edu. Primes.