NGUYỄN THỊ KIM TUYẾN

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
----------o0o----------

LUẬN VĂN THẠC SĨ KHOA HỌC

NGÀNH: CÔNG NGHỆ THÔNG TIN

NGHIÊN CỨU TỔNG QUAN VỀ TÍNH TOÁN LƯỚI VÀ CÀI ĐẶT MÔ HÌNH THỬ NGHIỆM

NGUYỄN THỊ KIM TUYẾN

HÀ NỘI 10-2006

NGUYỄN THỊ KIM TUYẾN CÔNG NGHỆ THÔNG TIN 2004-2006

HÀ NỘI 2006

LỜI CẢM ƠN

Trong quá trình tìm hiểu nghiên cứu để hoàn thành luận văn, tôi gặp không ít khó khăn, nhưng những lúc như vậy, tôi luôn nhận được sự động viên, khích lệ của thầy giáo, ***TS. Nguyễn Kim Khánh****. Thầy đã giúp đỡ tôi rất nhiều trong quá trình nghiên cứu, hướng dẫn tận tình trong cách thức và phương pháp nghiên cứu khoa học cũng như hỗ trợ tôi trong việc tìm tài liệu.*

Để có được những kết quả trong luận văn này, tôi xin gửi lời cảm ơn sâu sắc đến thầy giáo, ***TS. Nguyễn Kim Khánh*** *khoa Công nghệ thông tin trường ĐHBKHN.*

Tôi cũng xin gửi lời cảm ơn đến các thầy cô và các bạn trên Trung tâm tính toán hiệu năng cao, trường ĐHBKHN.

Cuối cùng tôi xin cảm ơn đồng nghiệp, gia đình và các bạn của tôi những người đã luôn bên cạnh, động viên và khích lệ tôi để có được kết quả như ngày hôm nay.

Hà Nội, ngày 05 tháng 10 năm 2006

Tác giả

Nguyễn Thị Kim Tuyến
Lớp Cao học CNTT 2004-2006

LỜI CAM ĐOAN

Tôi là Nguyễn Thị Kim Tuyến, học viên lớp cao học khoá 2004-2006, chuyên ngành Công nghệ thông tin. Tôi xin cam đoan bài luận văn "Nghiên cứu tổng quan về tính toán lưới và cài đặt mô hình thử nghiệm" là do tôi nghiên cứu, tìm hiểu dưới sự hướng dẫn của thầy giáo **TS. Nguyễn Kim Khánh,** không phải sự sao chép của người khác. Tôi xin chịu trách nhiệm về lời cam đoan này.

Hà Nội, ngày 05 tháng 10 năm 2006

Tác giả

Nguyễn Thị Kim Tuyến
Lớp Cao học CNTT 2004-2006

MỤC LỤC

DANH MỤC THUẬT NGỮ

Viết tắt	Tên đầy đủ	Chú giải
	Grid Computing	Tính toán lưới
	Globus Toolkit	Bộ công cụ middleware hỗ trợ tính toán lưới, cung cấp một số dịch vụ đệ trình công việc, quản lý tài nguyên, hạ tầng bảo mật, cũng như hỗ trợ việc xây dựng các dịch vụ lưới…
	Web Service	Dịch vụ web – một kiến trúc phát triển bởi W3C nhằm cung cấp các chức năng cho người dùng từ xa
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.
CAS	Community Authorization Service	Dịch vụ chứng thực cộng đồng. Một dịch vụ bảo mật trong môi trường lưới cho phép dung hòa giữa chính sách sử dụng tài nguyên của cộng đồng người dùng với chính sách sử dụng tài nguyên của những nhà cung cấp
DPSS	Distributed Parallel Storage System	Hệ thống lưu trữ song song phân tán: kỹ thuật tổ chức một tập các đĩa cứng nằm trên các server kết nối với nhau qua mạng diện rộng, cung cấp khả năng truy cập mức độ khối logic đến những bộ dữ liệu lớn.
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 and Management Service	Dịch vụ quản lý và định vị tài nguyên lưới
GTCP	Grid Telecontrol Protocol	Giao thức điều khiển lưới từ xa
FTP	File Transfer Protocol	Giao thức truyền tệp nổi tiếng qua mạng
Globus XIO	Globus eXtensible Input/Ouput	Giao diện vào ra mức thấp trong kiến trúc Globus
GMA	Grid Monitoring Architecture	Hệ thống gián sát lưới
GridFTP	Grid File Transfer Protocol	GridFTP là mở rộng của giao thức FTP, tích hợp khả năng bảo mật lưới, truyền dữ liệu tốt hơn so với FTP
GSI	Grid Security Infrastructure	Cơ sở hạ tầng bảo mật lưới trong kiến trúc của Globus, hỗ trợ giấy chứng nhận theo chuẩn X509 và dùng hệ mã công khai.
HPSS	High Performance Storage System	Hệ thống quản lý hiệu quả hàng trăm terabyte tới petabyte được lưu trên ổ cứng hoặc băng từ, liệu thường xuyên được sử dụng sẽ được lưu trên đĩa cứng, còn dữ liệu có tần suất sử dụng ít hơn sẽ được lưu trên băng từ.
HTTP	Hypertext Transfer Protocol	Giao thức truyền siêu văn bản, được sử dụng để truyền thông tin từ các máy phục vụ www đến các trình duyệt.
LDAP	Lightweight Directory Access Protocol	Giao thức đặc tả các kỹ thuật định danh đối tượng, mô hình dữ liệu, tìm kiếm và ghi các khoản mục dữ liệu.
LFN	Logical File Name	Tên logic của một thực thể dữ liệu trong lưới dữ liệu, hàm chứa nội dung của thực thể dữ liệu đó.
LRC	Local Replica Catalogue	Catalog định vị bản sao địa phương, lưu trữ tập các ánh xạ bao gồm hai trường: {tên logic của thực thể dữ liệu, vị trí vật lý cụ thể của thực thể đó}

MCS	Metadata Catalog Service	Dịch vụ siêu dữ liệu của kiến trúc lưới dữ liệu Globus, cho phép gắn các đối tượng dữ liệu với một số thuộc tính mô tả.
MDS	Monitoring and Discovery Service	Dịch vụ theo dõi và định vị tài nguyên
MPI	Message Passing Interface	Giao diện truyền thông điệp, cách thức trao đổi thông tin giữa các tiến trình.
OGSA	Open Grid Service Architecture	Kiến trúc dịch vụ lưới, định nghĩa các giao diện chuẩn và cơ chế hoạt động của dịch vụ lưới
OGSI	Open Grid Service Infrastructure	Hạ tầng dịch vụ lưới mở
PBS	Protable Batch System	Là hệ thống phân tải và quản lý tài nguyên rất mạnh. Được sử dụng phổ biến trong các hệ thống tính toán song song. Cung cấp khả năng khởi tạo và lập lịch cho việc thực thi và sắp xếp các công việc trên máy trạm.
PI	Protocol Interperter	Bộ thông dịch giao thức, có nhiệm vụ quản lý các kênh điều khiển trong kiến trúc GridFTP.
RFT	Reliable File Transfer Service	Dịch vụ truyền file tin cậy
RLI	Replica Location Index	Lưu các thông tin chỉ mục cho dịch vụ định vị bản sao, mỗi bản ghi bao gồm {LFN, và con trỏ tới LRC tương ứng}
RLS	Replica Location Service	Dịch vụ định vị bản sao trong kiến trúc lưới dữ liệu Globus, cho phép xác định vị trí của các bản sao của thực thể dữ liệu trong lưới.
RMI	Remote Method Invocation	Gọi phương thức từ xa, sử dụng trong Java khi chạy RMI, các đối tượng trong Java có thể gọi các phương thức của các đối tượng ở xa đang chạy trên một máy ảo khác.
RSL	Resource Specification Language	Ngôn ngữ đặc tả tài nguyên
SMTP	Simple Mail Transfer Protocol	Giao thức từ máy phục vụ đến máy phục vụ hỗ trợ phân tán thư điện tử
SOA	Service Oriented Architecture	Kiến trúc hướng dịch vụ
SOAP	Simple Object Access Protocol	Giao thức truy cập đối tượng từ xa đơn giản, dùng trong xây dựng các dịch vụ web, lập tình phân tán.
SDK	Software Development Kit	Tập công cụ hỗ trợ phát triển phần mềm
SRB	Storage Resource Broker	Bộ môi giới tài nguyên lưu trữ: thực hiện việc môi giới giữa các ứng dụng và các chủ tài nguyên lưu trữ, để xác định tài nguyên phù hợp nhất cho ứng dụng.
SSL	Secure Socket Layer	Giao thức bảo mật lưới
UHE	User Host Environment	Môi trường người dùng
VO	Virtual Organizations	Các tổ chức ảo
WSDD/ WSDL	Web Service Deployment Descriptor	Ngôn ngữ đặc tả dịch vụ web
WSRF	Web Service Protocol	Framework đưa ra bởi GT4 hỗ trợ kiến trúc lập trình mới.
XML	Extensible Markup Language	Là một cách thức linh động để tạo ra các định dạng thông tin và chia sẻ cả định dạng và dữ liệu trên web

DANH MỤC HÌNH

LỜI NÓI ĐẦU

Trong những năm gần đây rất nhiều thiết bị phần cứng mạnh phục vụ cho yêu cầu tính toán hiệu năng cao đã được tạo ra. Nhưng, do nhu cầu của con người là không giới hạn nên họ luôn thấy là chưa đủ, vì thể tính toán lưới đã ra đời nhằm đáp ứng nhu cầu này. Tuy nhiên điểm chính yếu của lưới không phải là sức mạnh tính toán mà là tính thực tiễn, tính thực tiễn này thể hiện ở chỗ các lưới tính toán thường được tạo ra dựa trên việc tận dụng các nguồn tài nguyên bình thường, sẵn có mà không cần phải mua hoặc tạo ra một hạ tầng phần cứng mới. Do đó, tính toán lưới nổi lên như một phương tiện tập hợp tài nguyên tính toán chi phí thấp để giải quyết những bài toán lớn.

Ở Việt Nam, công nghệ lưới còn khá mới, chỉ được triển khai ở một số ít các trung tâm tính toán tại viện nghiên cứu hoặc các trường đại học chuyên ngành lớn. Vì thế, luận văn đã được viết với mục đích nhằm nghiên cứu lý thuyết về tính toán lưới, hạ tầng cần thiết cho lưới, xây dựng một môi trường tính toán lưới phục vụ cho nhu cầu nghiên cứu các lĩnh vực chuyên sâu về lưới sau này. Luận văn được chia làm 5 chương với nội dung cụ thể như sau:

Chương 1. **Tổng quan về tính toán lưới**, trình bày những vấn đề chung nhất về tính toán lưới như định nghĩa, kiến trúc, các thành phần chính …

Chương 2. **Tổng quan về Globus**, trình bày chi tiết về thành phần nền tảng của tính toán lưới là bộ công cụ Globus Toolkit.

Chương 3. **Các kỹ thuật lưới hiện đang được triển khai ở nước ta,** trình bày các kỹ thuật lưới đang được một số trung tâm tính toán ở nước ta triển khai như Desktop Grid, Cluster Grid và Kết nối một Cluster vào một Grid thông qua PBS.

Chương 4. **Xây dựng lưới thử nghiệm**, trình bày các bước cơ bản để thiết kế một lưới. Cài đặt thử nghiệm một lưới đồng thời thực hiện kết nối một cluster vào lưới.

Phần **Kết luận**, trình bày tóm tắt kết quả đạt được và hướng phát triển tiếp theo của luận văn.

CHƯƠNG 1. TÍNH TOÁN LƯỚI

1.1 Tổng quan về Tính toán lưới

1.1.1 Tính toán lưới là gì?

➢ ***Định nghĩa***

Từ trước đến nay, mỗi tổ chức, cá nhân tùy theo cách quan niệm và thực tế xây dựng hệ thống của mình mà đưa ra những định nghĩa khác nhau về lưới.

Dưới quan điểm cá nhân của *I.Foster và các đồng nghiệp* thì "Một lưới là một hệ thống có các đặc trưng như tài nguyên được điều phối một cách phi tập trung; sử dụng các giao thức chuẩn, mở và đa năng; cung cấp chất lượng dịch vụ không tầm thường" - I. Foster‘s Three-Point Checklist (HPCWIRE - 22.07.2002)

Còn dưới quan điểm của một số công ty và liên minh phát triển lưới uy tín trên thế giới thì tính toán lưới được định nghĩa như sau [2]:

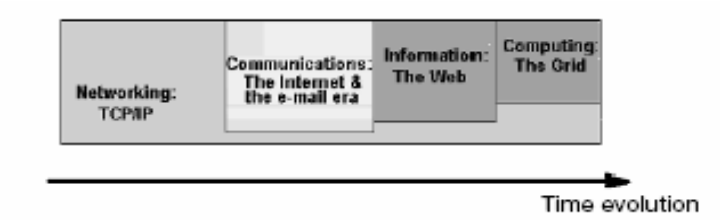
- *Định nghĩa của Oracle*: tính toán lưới là việc liên kết nhiều máy chủ và thiết bị lưu trữ thành một siêu máy tính nhằm tối ưu hóa được tính ưu việt của các hệ thống máy chủ cũng như hệ thống ứng dụng, nhờ đó giảm thiểu đến mức thấp nhất chi phí.
- *Định nghĩa của IBM*: tính toán lưới là một môi trường 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.
- *Định nghĩa của liên minh điện toán lưới*: môi trường tính toán lưới được hiểu như một hạ tầng kết nối hệ thống máy tính, hệ thống mạng, hệ thống cơ sở dữ liệu được sở hữu và quản lý bởi nhiều tổ chức, cá nhân nhằm cung cấp môi trường tính toán ảo duy nhất với hiệu năng cao cho người sử dụng.

Để có cái nhìn toàn diện về tính toán lưới, luận văn xin phép không đưa ra một định nghĩa cụ thể nào, thay vào đó chúng ta sẽ xem xét khái niệm tính toán lưới theo một số đặc điểm chung sau:

- *Kích thước lớn*: theo số lượng tài nguyên và khoảng cách địa lý giữa chúng
- *Phân tán*: có độ trễ đáng kể trong truyền dữ liệu, tài nguyên trải dài trên các vùng địa lý khác nhau
- *Động*: các tài nguyên có thể thay đổi khi ứng dụng đang được thực hiện
- *Hỗn tạp*: kiến trúc và tính chất của các nút lưới có thể là hoàn toàn khác nhau. Tài nguyên lưới có thể là các máy đơn hoặc mạng con khác nhau
- *Vượt qua phạm vi một tổ chức*: có nhiều trạm và các chính sách truy nhập có thể khác nhau trên các trạm, tổng thể lưới sẽ tạo ra một tổ chức ảo thống nhất
- Cơ chế và chính sách an toàn bảo mật phức tạp. Cơ chế quản lý tài nguyên đa dạng, phức tạp

➢ ***Lịch sử ra đời***

Quá trình ra đời của tính toán lưới có thể tóm tắt như sau:



Hình 1-1 Sự phát triển từ Networking đến Grid Computing

Cơ sở hạ tầng truyền thông của tính toán lưới là mạng Internet. Sau này WWW ra đời cuối những năm 80 đã tạo ra một cuộc cách mạng trong quá trình tính toán và chia sẻ thông tin trên mạng. Giao thức HTTP cùng với trình duyệt ngôn ngữ HTML đã cung cấp một cơ chế cho phép liên kết các văn bản và truy cập các thông tin trực tuyến, dễ dàng và ở bất kỳ đâu. Công nghệ XML được ra đời vào những năm 1994, đây là một chuẩn mới cho quá trình trao đổi thông tin trên mạng.

Công nghệ quang học đã cung cấp các dịch vụ và khả năng kết nối nhanh, rộng với chi phí hợp lý. Tốc độ mạng giữa những năm 1990 đã lên đến 2.4Bbps. Thông lượng lớn là nhân tố chính thúc đẩy hình thành công nghệ tính toán lưới.

Vào những năm 1990, khi các máy tính, máy trạm và mạng tốc độ cao cùng với các thiết bị hiện đại khác ra đời làm nảy sinh nhu cầu gộp nhóm các máy tính thành các cluster dùng cho công việc tính toán tốc độ cao. Sự xuất hiện của cluster

chính là mầm mống của hệ thống tính toán ngang hàng và các lưới tính toán sau này.

Lưới dữ liệu được bắt đầu năm 1999 với Globus Tookit 2.0+. Giai đoạn tiếp theo với sự ra đời của kiến trúc dịch vụ lưới mở rộng (OGSA) vào năm 2001 và sản phẩm Globus Tookit 3.0. Giai đoạn hiện tại tính từ những năm 2003 đến nay, được đánh giá bởi các cố gắng chuẩn hóa công nghệ và giao thức tính toán.

- ***Lợi ích của Tính toán lưới***

Các lợi ích mà tính toán lưới mang lại [2] bao gồm:

- Khả năng khai thác các tài nguyên nhàn rỗi
- Cung cấp khả năng xử lý song song
- Giúp hợp tác giữa các tổ chức
- Giúp truy nhập tài nguyên khác như đường truyền và các phần mềm đắt tiền
- Giúp cân bằng trong sử dụng tài nguyên
- Mang lại độ tin cậy cao

- ***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…

1.1.2 So sánh với các mô hình, công nghệ khác

Khi so với cluster, người ta thấy tầng trung gian của cả hai đều đưa ra cơ chế truyền thông điệp cho các ứng dụng song song. Như vậy, về bản chất kiến trúc mức cao của một cluster là tương tự như của một lưới. Tuy nhiên giữa chúng vẫn có những điểm khác biệt quan trọng như:

- Về mặt phạm vi, Clusters được đặt trong một phòng hoặc một toà nhà còn Grids lại có thể được đặt phân tán trên trên các vùng địa lý khác nhau
- Clusters có một hệ quản trị duy nhất, ngược lại Grids có các biên quản trị kéo dài

- Clusters chỉ chú tâm đến các bài toán chuyên về tính toán hiệu năng. Còn Grids giải quyết các bài toán tính toán phân tán và chia sẻ tài nguyên
- Clusters thường bao gồm các hệ thống con đồng nhất còn Grids lại là sự kết hợp của nhiều dạng hệ thống con khác nhau
- Clusters thường có lượng tài nguyên cố định nhưng tài nguyên trên Grids thay đổi theo yêu cầu thực tế và khả năng dùng được của tài nguyên

Tuy có điểm khác biệt nhưng vẫn có thể coi tính toán phân cụm như một trường hợp đặc biệt của tính toán lưới. Trên thực tế Grid không phải là một cuộc cách mạng mới mà thực chất nó là bước tiến trong công nghệ điện toán phân tán. Giống như Web, tính toán lưới giảm bớt độ phức tạp khi nhiều người cùng khai thác một nền hoạt động thống nhất nó chỉ khác ở sự hỗ trợ liên lạc. Khi so với mạng ngang hàng (P2P) thì tính toán lưới cũng cho phép người sử dụng chia sẻ file và nhiều loại tài nguyên khác nữa. Nói chung, với các công nghệ ảo khác, grid giống ở chỗ cho phép ảo hóa các nguồn nhân lực CNTT nhưng khác biệt là grid cho phép ảo hóa những nguồn tài nguyên tản mát và vô cùng rộng lớn.

1.1.3 Phân loại mạng lưới

Lưới thường được cài đặt ở nhiều dạng khác theo ứng dụng cụ thể hoặc theo cấu trúc của tổ chức ảo tham gia lưới hoặc theo tính chất của tài nguyên được chia sẻ. Sau đây là một số dạng lưới [3]:

- ***Departmental Grids***
  - *Cluster Grids*: gồm một hoặc nhiều hệ thống kết hợp lại nhằm cung cấp một điểm truy xuất đơn cho người dùng. Nó phù hợp cho các ứng dụng yêu cầu hiệu năng tính toán và băng thông lớn
  - *Infra Grids*: do IBM đề xuất nhằm định nghĩa một lưới tối ưu việc sử dụng tài nguyên trong một xí nghiệp
- ***Enterprise Grids***
  - *Enterprise Grids*: được triển khai trong các công ty lớn có chi nhánh ở nhiều nơi trên thế giới có nhu cầu chia sẽ tài nguyên

- *Intra Grids*: tài nguyên chia sẻ trong các nhóm khác nhau của một xí nghiệp tạo thành một intra grid
- *Campus Grids*: cho phép nhiều dự án hoặc nhiều bộ phận chia sẻ tài nguyên tính toán theo hướng cộng tác.

➢ ***Extraprise Grids***

- *Extra Grids*: cho phép chia sẻ tài nguyên với các đối tác bên ngoài. Liên kết giữa các tổ chức này được thiết lập bằng các dịch vụ tin cậy.
- *Partner Grids*: là các mạng lưới giữa các tổ chức, công ty, xí nghiệp trong cùng lĩnh vực mà có nhu cầu cộng tác nhằm thực hiện dự án chung

➢ ***Global Grids***

- *Global Grids*: cho phép người dùng khai thác các tài nguyên bên ngoài. Nó cung cấp sức mạnh của các tài nguyên phân tán trên khắp thế giới.
- *Inter Grids*: cung cấp khả năng chia sẻ tài nguyên và lưu trữ thông qua Web

➢ ***Compute Grids***

Compute grids được thành lập nhằm mục tiêu chia sẻ tài nguyên tính toán.

- *Desktop Grids*: đây là mạng lưới tập hợp sức mạnh tính toán của các máy tính để bàn. Các mạng lưới thuộc dạng này chủ yếu chạy trên nền Windows
- *Server Grids*: một số tập đoàn lớn thường có các tài nguyên máy chủ được quản lý bởi bộ phận IT nằm tại nhiều địa điểm khác nhau. Họ muốn tạo Server Grids để chia sẻ những tài nguyên đắt giá này
- *High-Performance/Cluster Grids*: mạng lưới gồm các hệ thống dành riêng cho tính toán như các siêu máy tính hoặc nhóm máy tính hiệu năng cao

➢ ***Data Grids***

Tài nguyên chủ yếu được chia sẻ trong mạng lưới này là dữ liệu. Nó được xây dựng nhằm tối ưu các thao tác hướng dữ liệu như lưu trữ, truy xuất, trao đổi.

➢ ***Utility Grids***

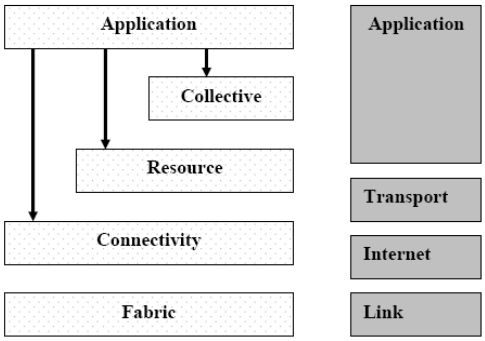
Utility grids là mạng lưới bao gồm các tài nguyên tính toán thương mại mà được duy trì, quản lý bởi các nhà cung cấp dịch vụ.

- *Service Grids*: cung cấp truy xuất tới các tài nguyên mà có thể được mua bởi một tổ chức

1.2 Kiến trúc và thành phần chính của hệ thống lưới

1.2.1 Tổng quan kiến trúc lưới

Lưới được xây dựng trên nền tảng kiến trúc mở và phân tầng. Trong mỗi tầng của lưới, các thành phần chia sẻ những thuộc tính chung và được bổ sung những tính năng mới mà không làm ảnh hưởng đến các tầng khác [4]. Ta có thể tổng hợp kiến trúc lưới thành 5 tầng như sau:



Hình 1-2 Kiến trúc phân tầng lưới

- ***Tầng chế tác (Fabric layer)***

Bao gồm các tài nguyên cục bộ phân tán trên mạng, chúng bị ràng buộc bởi *Cơ chế quản lý tài nguyên* và *Cơ chế thẩm tra*. Người ta phân tài nguyền của tầng này thành các nhóm chính sau:

- *Tài nguyên tính toán*: cho phép kiểm soát, điều khiển việc thực thi công việc
- *Tài nguyên lưu trữ*: dùng để lấy về/tải lên các tập tin, cho phép đọc một phần tập tin hoặc chọn lọc dữ liệu từ tập tin ở xa
- *Tài nguyên mạng*: là môi trường mạng truyền thông
- *Các kho mã nguồn*: là nơi quản lý tất cả các loại tài nguyên và các phiên bản của mã nguồn

- ***Tầng kết nối (Connectivity layer)***

Tầng này đóng vai trò rất quan trọng, nó gồm các giao thức xác thực và truyền thông. *Truyền thông* bao gồm việc truyền thông tin, định tuyến và đặt tên. Những giao thức này tương tự các giao thức IP, TCP, UDP trong bộ giao thức TCP/IP và các giao thức tầng ứng dụng như DNS, OSPF, RSVP…*Vấn đề bảo mật* được giải quyết bằng giải pháp xác thực như:

- *Cơ chế đăng nhập một lần (Single Sign On)*: cho phép người dùng chỉ cần đăng nhập vào mạng lưới một lần duy nhất cho tất cả các truy cập các tài nguyên được phép trong tầng chế tác cho đến khi kết thúc đăng nhập
- *Cơ chế ủy quyền (Delegation, Proxy)*: người dùng có thể ủy quyền truy cập tài nguyên hợp pháp lại cho một chương trình trong một khoảng thời gian xác định. Chương trình này cũng có thể ủy quyền có điều kiện một phần các tập quyền của nó cho chương trình con khác
- *Cơ chế tích hợp đa giải pháp bảo mật địa phương (Integration with various local security solutions)*: cơ chế bảo mật mạng lưới phải có khả năng giao tiếp trong với các cơ chế bảo mật địa phương mà không yêu cầu thay thế toàn bộ các giải pháp bảo mật hiện có, nhưng cần có cơ chế ánh xạ bảo mật trong các môi trường cục bộ khác nhau
- *Cơ chế quan hệ tin tưởng dựa trên người dùng (User-based Trust Relationships)*: người dùng có thể sử dụng các loại tài nguyên có được từ sự kết hợp của nhiều nhà cung cấp khác nhau

➢ ***Tầng tài nguyên (Resource layer)***

Tầng này được xây dựng trên nền tảng sẵn có của tầng kết nối. Những giao thức trong tầng tài này sẽ gọi các chức năng trong tầng chế tác để truy cập và sử dụng các loại tài nguyên cục bộ. Nó có hai loại giao thức chính:

- *Giao thức thông tin (Information protocol)*: cho phép lấy các thông tin về cấu trúc, tình trạng của một loại tài nguyên nào đó trong mạng lưới
- *Giao thức quản lý (Management protocol)*: dùng để sắp xếp quản lý thứ tự các truy cập đến các tài nguyên được chia sẻ

➢ ***Tầng kết hợp (Collective layer)***

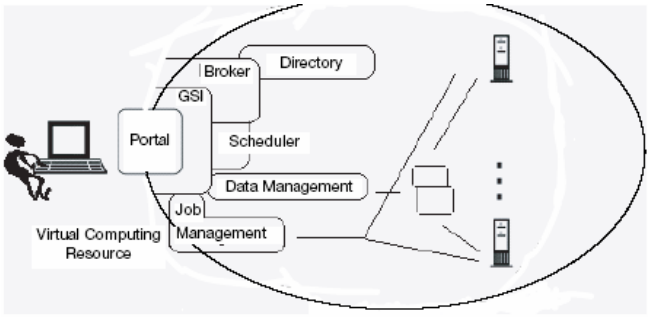
Trong khi tầng tài nguyên chỉ cho phép truy cập đến một loại tài nguyên đơn thì tầng kết hợp lại chứa các giao thức và dịch vụ cho phép giao tiếp giữa các tài nguyên trong mạng lưới. Tầng này bao gồm các dịch vụ chính như sau:

- *Các dịch vụ thư mục (Directory Services)*: cho phép tìm hiểu sự tồn tại cũng như thuộc tính của các nguyên như loại tài nguyên, tính khả dụng …
- *Các dịch vụ cấp phát chung, lập lịch, môi giới (Co-allocation, Scheduling & Broker Services)*: cho phép gửi yêu cầu đến một hay nhiều máy chủ cho mục đích cấp phát, lập lịch và môi giới tài nguyên tương ứng
- *Các dịch vụ giám sát và dự báo (Monitoring and Diagnostic Services)*: cho phép hệ thống hỗ trợ kiểm soát tài nguyên trong lưới
- *Các dịch vụ nhân bản dữ liệu (Data Replication Services)*: hỗ trợ việc quản lý lưu trữ tài nguyên, giúp việc truy cập tài nguyên lưới trở nên dễ dàng hơn
- *Các hệ hỗ thống trợ lập trình lưới (Grid-enable Programming Systems)*: gồm các thư viện lập trình
- *Hệ thống quản lý tải và môi trường cộng tác (Workload Management System & Collaboration Framework)*: cung cấp các đặc tả, quản lý tính đồng bộ, đa luồng, đa thành phần… trong các tiến trình tính toán
- *Dịch vụ tìm kiến phần mềm (Software Discovery Service)*: hỗ trợ tìm kiếm và lựa chọn phần mềm cài đặt và làm nền tảng cho mạng lưới

➢ ***Tầng ứng dụng (Application layer)***

Tầng này bao gồm các ứng dụng được phát triển trên môi trường lưới như: các ứng dụng sinh học, vậy lý, thiên văn, tài chính… Về nguyên tắc, người sử dụng có thể tương tác với lưới thông qua tầng ứng dụng một cách trong suốt mà không nhận biết được sự có mặt của các tầng khác trong lưới.

1.2.2 Các thành phần theo mô hình chức năng



Hình 1-3 Các thành phần theo mô hình chức năng

Nhìn từ hình trên ta thấy về mặt chức năng thì lưới gồm các thành phần sau:

1. Cổng tương tác (Grid portal): là một giao diện cho phép người dùng sử dụng các ứng dụng lưới, do đó lưới trở nên trong suốt với người dùng.

2. Thành phần bảo mật (Security): là cơ chế đảm bảo các hoạt động như xác thực, cấp quyền, bảo mật-toàn vẹn dữ liệu và tính sẵn sàng của dữ liệu.

3. Chức năng an ninh nút (Node Security Function): chức năng này chịu tránh nhiệm xác thực và bảo mật cho từng nút trong quá trình giao tiếp giữa nó và các thành phần khác bên trong mạng lưới. Nó phụ thuộc vào hệ điều hành và các hệ thống lưới cụ thể, thường thấy là cơ chế cấp chứng chỉ quyền truy cập.

4. Bộ lập lịch (Scheduler): là phần phối hợp quá trình thực thi của nhiều công việc song song. Đơn giản, người sử dụng có thể chọn nút thích hợp để chạy tác vụ, sau đó chỉ việc kích hoạt lệnh để định tuyến công việc đó tới nút đã chọn.

5. Thành phần môi giới (Broker): sau khi người dùng được xác nhận quyền gia nhập vào mạng lưới bởi thành phần an ninh nút, thành phần này sẽ chỉ rõ ứng dụng của người dùng được sử dụng tài nguyên nào và đảm bảo tài nguyên được sẵn sàng sử dụng theo tham số truyền vào.

6. Quản lý, phân bổ tài nguyên (grid resource allocation manager, GRAM): cung cấp dịch vụ để kích hoạt từng công việc trên từng tài nguyên cụ thể; kiểm tra trạng thái công việc; đọc kết quả khi công việc đó kết thúc. Các thông tin của thành phần này sau đó sẽ được bộ lập lịch sử dụng.

7. Tài nguyên (Resource): tài nguyên lưới bao gồm bộ xử lý, bộ lưu trữ, các ứng dụng và các thành phần.

8. Quản lý dữ liệu (Data management): dữ liệu có thể nằm ở tài nguyên, hoặc là kết quả thực thi của một tác vụ nào đó. Thành phần quản lý dữ liệu phải đảm bảo an toàn và ổn định trong quá trình di chuyển dữ liệu giữa các lưới.

9. Giao thức (Protocol): là thành phần đảm bảo liên kết các thành phần chức năng kể trên để có thể hoạt động và tương tác được với nhau trong mạng lưới.

1.2.3 Các thành phần theo mô hình vật lý

Dựa trên tài liệu [3], các thành phần của lưới theo mô hình vật lý bao gồm:

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 đên hiệu suất lưới.

2. Thành phần tính toán (Computation): được cung cấp bởi các bộ xử lý trong lưới, chúng đa dạng về tốc độ, kiến trúc, nền tảng phần mềm và lưu trữ.

3. Thành phần lưu trữ (Storage): dữ liệu có thể được lưu trữ phân tán trên nhiều thiết bộ xử lý hoặc một mạng SAN. 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

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 vất đề cần được quan tâm nhất.

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.

1.3. Các chuẩn cho tính toán lưới

Lưới tính toán thường gồm một tập các tài nguyên không đồng nhất. Một ứng dụng lưới thường có nhiều thành phần, dịch vụ khác nhau. Đồng thời các dịch vụ này lại thường xuyên tương tác với nhau. Càng nhiều dịch vụ thì số tương tác giữa chúng càng tăng và rất dễ dẫn đến tình trạng hỗn loạn. Nếu mỗi dịch vụ sử dụng một cách riêng để tương tác với các dịch vụ khác thì vấn đề giao tiếp giữa các

dịch vụ lưới sẽ rất phức tạp. Do đó, cần thiết là phải có chuẩn định nghĩa giao diện giao tiếp chung cho các dịch vụ này.

1.3.1 OGSA/OGSI là gì?

OGSA và OGSI [2] được Global Grid Forum (GGF) phát triển cho mục tiêu chuẩn hoá. GGF định nghĩa các chuẩn mạng lưới trong phạm vi các trình ứng dụng, các mô hình lập trình, quản trị dữ liệu, bảo mật, thực thi, lập lịch và quản lý tài nguyên.

OGSA (Open Grid Services Architecture): định nghĩa một chuẩn kiến trúc mới cho các ứng dụng chạy trên lưới. OGSA định nghĩa dịch vụ lưới là gì, chúng có khả năng gì, và dựa trên nền công nghệ nào. Nhưng OGSA không đưa ra đặc tả chi tiết và kỹ thuật cần để triển khai một dịch vụ lưới.

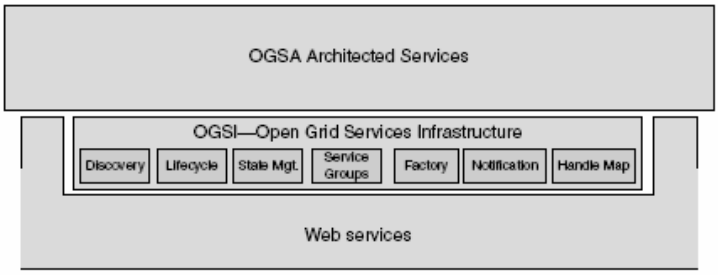
OGSI (Open Grid Services Infrastructure): nặng về đặc tả kỹ thuật cho các khái niệm được đưa ra trong OGSA. OGSI định nghĩa các cơ chế tạo mới, quản trị và trao đổi thông tin giữa các dịch vụ lưới. Một dịch vụ lưới là dịch vụ web thích ứng với tập hợp các quy ước về giao diện và cách đáp ứng để xác định cách một client tương tác với một lưới. Đặc biệt, OGSI còn định nghĩa các giao diện chuẩn và qui tắc của dịch vụ lưới – xây dựng trên cơ sở các dịch vụ web.

Globus Toolkit là phần triển khai của OGSI, nó rất hữu dụng trong việc triển khai những gì được đặc tả bởi.

Grid Services là các dịch vụ lưới dựa trên nền tảng Web Services và được mở rộng từ Web Services.

1.3.2 Chuẩn OGSI

OGSI giới thiệu một mô hình tương tác cho các dịch vụ lưới. Mô hình này cung cấp một phương thức cố định bằng cách đưa ra các giao diện dùng trong phát hiện, quản lý vòng đời, trạng thái, tạo-huỷ, thông báo sự kiện và quản lý tham chiếu.



Hình 1-4 Mối quan hệ giữa OGSA và OGSI

1. Factory: là cơ chế (giao diện) cung cấp cách tạo các dịch vụ lưới mới. Các Factory có thể tạo ra nhiều thể hiện tạm của một chức năng hạn chế, chẳng hạn một bộ lập lịch tạo một dịch vụ để mô tả cách thực hiện một công việc thông thường; hoặc chúng có thể tạo ra các dịch vụ tồn tại trong thời gian dài như việc nhân bản cục bộ một tập dữ liệu được sử dụng liên tục.

2. *Life cycle*: là cơ chế dùng để ngăn các dịch vụ lưới truy cập đến các tài nguyên không được yêu cầu. Các dịch vụ lưới được tạo với vòng đời xác định.

3. *State managemen*: mọi dịch vụ lưới đều có một trạng thái. OGSI xác định một khung để biểu diễn các trạng thái và một cơ chế để kiểm duyệt hoặc sửa đổi chúng. OGSI cũng quy định số các trạng thái tối thiểu mà mỗi dịch vụ lưới phải có.

4. *Service groups*: là tập các dịch vụ lưới được chỉ định cho một mục đích riêng nào đó.

5. *Notification*: các dịch vụ tương tác với nhau thông qua cơ chế trao đổi thông điệp trên các lời triệu gọi dịch vụ. Thông tin trạng thái được mô hình cho các dịch vụ lưới sẽ thay đổi khi hệ thống chạy.

6. *Handle Map*: dùng cho các vấn đề nhận dạng. Khi các factory được sử dụng để tạo ra một thể hiện mới của dịch vụ lưới, Factory trả về định danh của thể hiện mới này. Định danh này là sự kết hợp của Grid Service Handle (GSH) và một Grid Service Reference (GSR). GSH cung cấp tham chiếu đến định danh của dịch vụ lưới còn GSR có thể thay đổi theo thời gian sống của các dịch vụ lưới.

- ***Các cài đặt của OGSI***

Đóng vai trò là phần cốt lõi của kiến trúc dịch vụ lưới, OGSI cần phải được triển khai trên một nền có hỗ trợ các dịch vụ Web. Thường thì nhà cung cấp sẽ cài đặt OGSI bằng cách sử dụng trực tiếp các thành phần thực thi mã mở có sẵn của Globus và sau đó tích hợp nó với platform của riêng họ như Webspere, Weblogic...

- ***Một số giao thức chuẩn***

SOAP (*Simple Object Access Protocol)*: là cơ chế truyền thông độc lập với cấu hình nền và các giao thức khác. Nó dùng XML để trao đổi thông tin trong môi trường phân tán. Nó hỗ trợ nhiều kiểu trao đổi thông tin khác nhau bao như: mô hình trao đổi thông tin sau khi gọi thủ tục từ xa hoặc mô hình trao đổi thông tin trong một cơ chế hướng thông báo. SOAP thường độc lập về giao thức, độc lập về ngôn ngữ, độc lập về cấu hình và hệ điều hành, hỗ trợ các thông báo XML SOAP.

UDDI (*Universal Description Discoverry and Intergation*): kho chứa phần lưu trữ của các dịch vụ Web. UDDI là phần định dạng mô tả dịch vụ Web chuẩn. Một đăng ký UDDI có thể chứa siêu dữ liệu cho bất kỳ kiểu dịch vụ nào. UDDI cho phép tìm kiếm và khai thác các dịch vụ Web hiệu quả hơn.

WSDL *(Web Service Definition Language)*: ngôn ngữ định nghĩa các dịch vụ web. Là ngôn ngữ cung cấp cách mô tả giao diện đặc biệt của các dịch vụ web và các API. WSDL có thể được xem như là tài liệu được viết bằng XML.

XML *(eXtensible Markup Language)*: ngôn ngữ đánh dấu mở rộng. Là một siêu ngôn ngữ, sử dụng để mô tả ngữ nghĩa của đối tượng và mô tả phần cấu trúc bên trong một kiểu mở.

WSIL *(WS-Inspection Language)*: tận dụng định dạng của XML làm tăng khả năng khai thác và tập hợp những mô tả của dịch vụ web đơn giản và mở rộng hơn. WSIL là cơ chế đơn giản để khai thác dịch vụ web, nó là một định dạng XML.

1.3.3 Chuẩn OGSA

OGSA xác định phạm vi các định dạng dịch vụ quan trọng được yêu cầu để hỗ trợ cho hệ thống mạng lưới và các ứng dụng trong lĩnh vực thương mại điện tử và khoa học. Nó xác định tập các dịch vụ nền tảng thiết yếu cho cho rất nhiều trình

ứng dụng và các hệ thống; xác định các chức năng được yêu cầu ở mức cao đối với mối quan hệ tương tác giữa các dịch vụ lõi.

Mục tiêu của OGSA là định nghĩa các phương thức và cơ chế chuẩn cho vấn đề cơ bản chung của các hệ thống lưới, chẳng hạn như cách giao tiếp giữa các dịch vụ, thiết lập định danh, định quyền truy cập, phát hiện tài nguyên-dịch vụ, thông báo lỗi, và quản lý tập các dịch vụ…

- ***Các thành phần cơ bản của OGSA***

Có ba thành phần chính của OGSA:

- *Cơ sở hạ tầng dịch vụ lưới mở OGSI*: xây dựng trên các kỹ thuật dịch vụ web và lưới, OGSI định nghĩa cơ chế tạo, quản lý và chuyển đổi thông tin giữa các dịch vụ lưới.
- *Các dịch vụ OGSA*: xây dựng trên các cơ chế OGSI để định nghĩa các giao diện và các hành vi kết hợp cho các chức năng không được hỗ trợ trực tiếp bởi OGSI như phát hiện dịch vụ, truy xuất dữ liệu, tích hợp dữ liệu…
- *Các mô hình OGSA*: hỗ trợ các đặc tả giao diện bằng cách định nghĩa các mô hình cho các tài nguyên chung và các kiểu dịch vụ.

- ***Các dịch vụ nền (Platform services)***

OGSA dùng thuật ngữ *platform services* để chỉ những dịch vụ cung cấp các chức năng cơ bản. Platform services (i) cung cấp các chức năng nền dùng để xây dựng các dịch vụ lưới khác (ii) cung cấp các chức năng chung dùng trong một số các dịch vụ mức cao (iii) cung cấp các chức năng được thiết kế để dùng cho các quan hệ mở rộng. Một chức năng được cung cấp bởi một dịch vụ nền sẽ được mô tả trong một số các dịch vụ mức cao. Hiện tại, tập các dịch vụ nền của OGSA gồm:

- *OGSI*: định nghĩa các dịch vụ lưới và các cơ chế nền để tạo, quản lý và trao đổi thông tin giữa các dịch vụ
- *WS-Agreement*: cung cấp một tập giao diện hỗ trợ việc điều chỉnh các chính sách, các thoả thuận mức dịch vụ, đặt trước …
- *CMM* (Common Management Model): cung cấp một cơ sở hạ tầng có thể quản lý được cho các tài nguyên trong OGSA. CMM định nghĩa mô hình cư

xử cơ sở cho tất cả các tài nguyên và các bộ quản lý tài nguyên trong lưới, công thêm chức năng quản lý các mối quan hệ và quản lý vòng đời

- OGSA Data Services (các dịch vụ dữ liệu OGSA): cung cấp các chức năng cơ bản để quản lý dữ liệu trong một môi trường lưới

➢ ***Các yêu cầu về mặt chức năng***

- *Yêu cầu chức năng cơ bản*: khám phá và môi giới; đo đạc và tính toán; chia sẻ dữ liệu; triển khai; tổ chức ảo; giám sát; chính sách
- *Yêu cầu bảo mật*: bảo mật đa phần; giải pháp bảo mật phạm vi; xác thực, uỷ quyền và cấp quyền; mã hoá; chứng thực
- *Yêu cầu về quản lý tài nguyên*: đồng nhất cách cung cấp, ảo hoá tài nguyên, tối ưu việc sử dụng, có khả năng lập lịch và cung cấp băng thông động, có khả năng truy cập theo lô và truy cập tương tác, hỗ trợ quản lý và giám sát việc sử dụng, lập lịch động cho các tác vụ, đảm bảo các tài nguyên được sử dụng như nhau, có khả năng đặt trước tài nguyên, có cơ chế ghi lại các xử lý, và phải quản lý được luồng công việc và phải định giá được việc sử dụng tài nguyên để lập hoá đơn cho người dùng
- *Các yêu cầu về đặc tính của hệ thống*: phải có khả năng chịu lỗi, phát hiện được hiểm hoạ, tự "chăm sóc sức khỏe" của tài nguyên, giám sát, theo dõi sự tấn công, quấy rầy, quản lý được các ứng dụng kế thừa, có thể "hệ thống hoá" và "tự động hoá" các hoạt động chuẩn cho bộ quản trị, có khả nằng khởi tạo yêu cầu tương tác theo thoả thuận giữa client và server và tạo nhóm/tập các dịch vụ, cho phép một số dịch vụ được kế thừa và sử dụng lại các dịch vụ đã tồn tại

1.4 Các thành phần chính trong mô hình chức năng của lưới

1.4.1 Bảo mật

Bảo mật luôn là một thành phần quan trọng trong bất kì 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 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 hoặc không giao tiếp, truy cập với các dữ liệu cá nhân.

1.4.1.1 Cơ chế bảo mật trong môi trường 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 mật 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 mật 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.

Formatted: Bullets and Numbering

1.4.1.2 Các chính sách bảo mật trong môi trường lưới

Sau đây là các chính sách bảo mật:

- *Môi trường lưới bảo mật đa miền*: tập trung điều khiển các tương tác liên miền, ánh xạ hoạt động liên miền với các chính sách bảo mật địa phương
- *Hoạt động lưới hạn chế trong đơn miền quản trị*: các hoạt động đa miền phải tuân theo chính sách bảo mật 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* 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

Formatted: Bullets and Numbering

1.4.1.3 Hạ tầng an ninh mạng lưới GSI (Grid Security Infrastructure)

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 uỷ quyền. GSI dựa trên các công nghệ mã khoá công khai (Public Key Infrastructure hay PKI), Chứng thư X.509 (Certificate), Nghi thức truyền thông bảo mật (Secure Socket Layer hay SSL).

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

Formatted: Bullets and Numbering

➢ ***Cơ sở hạ tầng khóa công khai***

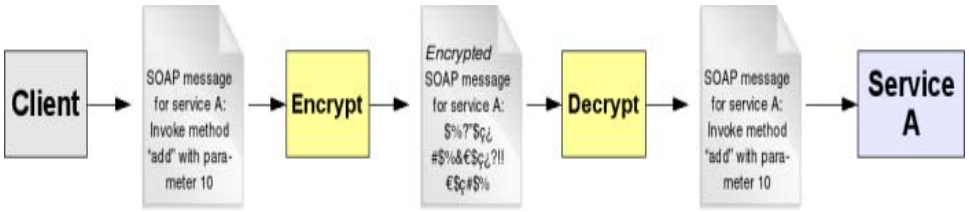
Chứng thực trong GSI là thao tác cung cấp cho mỗi thực thể một tên định danh duy nhất bằng cách đưa ra khái niệm giấy ủy quyền lưới, nó là một cặp giấy chứng nhận và khóa mã hóa (khóa bí mật). Trong môi trường PKI, mỗi thực thể phải trao quyền sở hữu khóa bí mật của mình để bảo đảm sự toàn vẹn của hệ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

Formatted: Bullets and Numbering

➢ ***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 1-5 Bảo mật mức giao vận



Hình 1-6 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***

Formatted: Bullets and Numbering

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.

➢ ***Sự ủy quyền***

Formatted: Bullets and Numbering

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.

➢ ***Chứng thực***

Formatted: Bullets and Numbering

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,...

1.4.2 Quản lý tài nguyên lưới

1.4.2.1 Những thách thức trong quản lý tài nguyên lưới

➢ ***Xuất phát từ đặc trưng của tài nguyên lưới***

Lưới không giữ quyền điều khiển tuyệt đối đối với 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 sự không đồng nhất, các tổ chức khác nhau có các chính sách quản lý tài nguyên khác nhau, 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 nữa đối với quản lý tài nguyên là vấn đề bảo mật 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 về bảo mật khác. Một lưới có đa dạng nguồn tài nguyên, mỗi tài nguyên lại yêu cầu các mức bảo mật khác nhau.

- ***Định vị tài nguyên lưới***

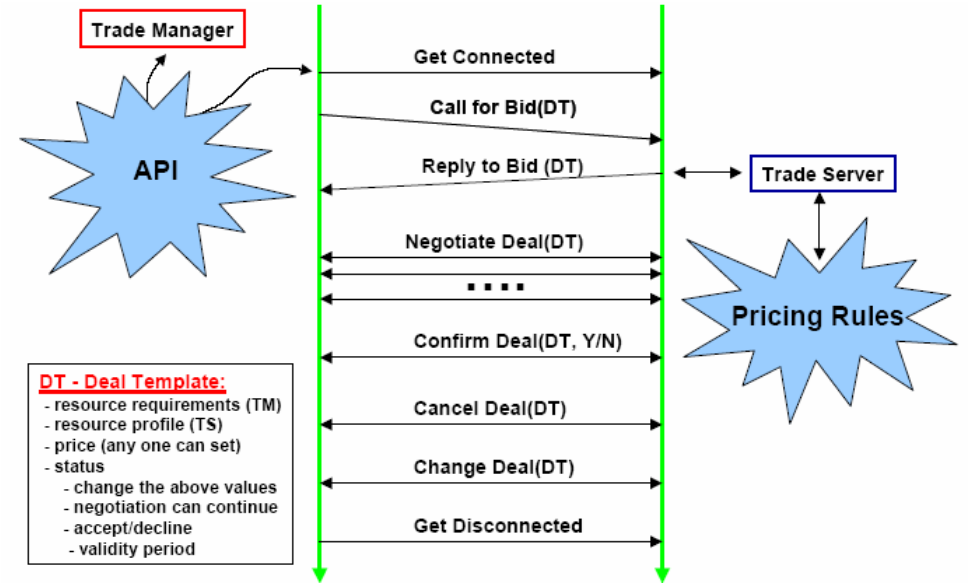
Khi có yêu cầu của người dùng, bộ quản lý tài nguyên sẽ tìm tài nguyên từ dịch vụ chỉ mục (Index Service) sau đó định vị tài nguyên đến một số nút cụ thể nào đó trong lưới và tại các nút này thì tài nguyên sẽ được lập lịch sử dụng. Khi một ứng dụng đang chạy, bộ quản lý tài nguyên cần theo dõi trạng 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.

- ***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 1-7 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 1-7 Mô hình thương lượng tài nguyên lưới

1.4.2.2 Hệ quản trị tài nguyên GRAM

GRAM là dịch vụ được xây dựng trên cơ chế bảo mật GSI, nó đóng vai trò là bộ quản lý, phân chia tài nguyên trong toàn bộ hệ thống tính toán lưới.

➢ ***Kiến trúc của GRAM***

Kiến trúc bên ngoài:

Để có thể đệ trình một công việc lên một host, người dùng sẽ thông qua các API 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 Gatekeeper tương ứng. Gatekeeper này sẽ xác thực những thông tin được gửi đến dựa vào cơ chế bảo mật GSI. Nếu tất cả đều hợp lệ, Gatekeeper sẽ tạo ra một Jobmanager để phục vụ cho job. Job manager sẽ phân tích kịch bản RSL 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, jobmanager 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. 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.

Kiến trúc bên trong:

Để có thể thực thi một công việc từ xa, GRAM Gatekeeper phải được chạy trên một remote computer, lắng nghe ở một cổng được quy định trước; công việc sẽ được thực thi trên remote computer đó. Việc thực thi bắt đầu khi ứng dụng người dùng chạy trên máy cục bộ gửi job request đến remote computer. Request 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 remote computer. Request job sẽ được xử lý bởi GRAM Gatekeeper, từ đó nó sẽ tạo ra một job manager tương ứng mà công việc yêu cầu. Lúc đó, Jobmanager sẽ theo dõi tình trạng thực thi job và chịu trách nhiệm thông báo thông tin của job cho người sử dụng.

1.4.3 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.

1.4.3.1 Giao thức truyền tập tin mạng lưới GridFTP

GridFTP là giao thức truyền tập tin giống như FTP [29] 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.

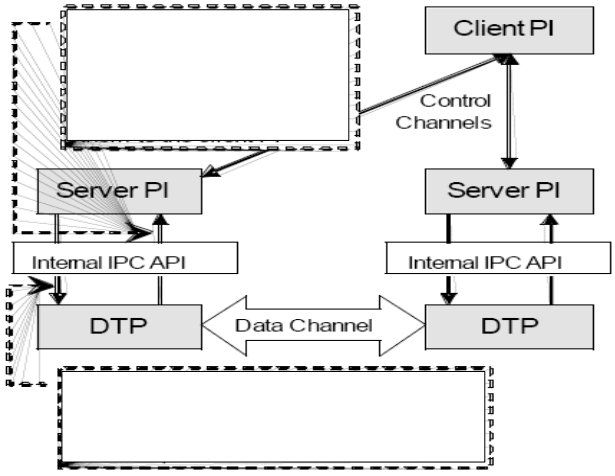
- Trao đổi dữ liệu với sự tham gia của phía thứ ba. Đây là nghi thức cho phép chuyển tập tin trực tiếp từ máy chủ tới máy chủ khi kênh điều khiển nằm trên máy chủ thứ ba
- Xác thực các kênh truyền dữ liệu
- Tái sử dụng các kênh truyền dữ liệu và dẫn truyền các lệnh điều khiển

➢ ***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:

- Cơ sở hạ tầng bảo mật lưới và hỗ trợ Kerberos
- 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 1-8 Kiến trúc của GridFTP

Bộ thông dịch giao thức PI:

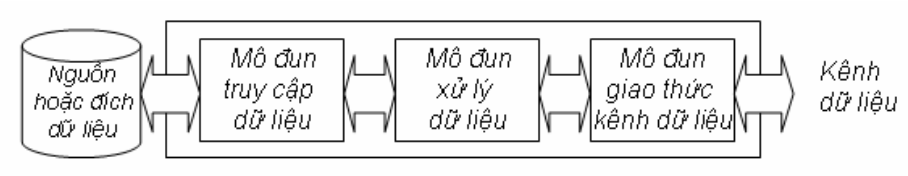
Server PI có nhiệm vụ quản lý kênh điều khiển, trao đổi thông tin với máy khách qua kênh này. Để máy khách liên lạc được với máy chủ GridFTP, server PI

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, và 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 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 1-9 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 cài trên máy khách, thực hiện gửi yêu cầu tới máy chủ GridFTPđể 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.

1.4.3.2 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 tất cả bản sao
- *Tính tin cậy*: lỗi xảy ra ở một miền không làm 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 như:

- *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, Cho một PFN, tìm tập LFN tương ứng với PFN đó
- *Về tính toàn vẹn cục bộ*: quản lý tính toàn vẹn giữa nội dung của tên lôgic 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, P_L, P_R, R, S, C). Bốn tham số đầu tiên (G, P_L, 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. Phần tiếp theo phân tích ý nghĩa từng tham số:

G: Số lượng RLI trong hệ thống

$P_{L:}$ Đặc trưng cho kiểu phân nhóm tên tệp lôgic trong RLI

P_R: 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 lôgic 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

1.4.4 Lập lịch trong môi trường lưới

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 [16]:

- 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.***

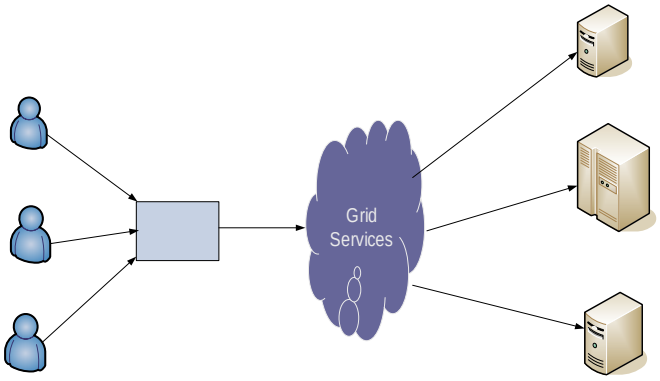
Tiến hành các bước để thực thi công việc trên tài nguyên đã chọn, giám sát trạng thái công việc và gửi kết quả lại cho người sử dụng

- *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

1.4.5 Grid Portal

Hệ thống TTL 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.



Hình 1-10 Cổng điện tử GridPortal

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 cho ứng 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 đế các site phục vụ cho một tác vụ truyền thông nào .

Client

Grid Portal

Để triển khai công nghệ GridPortal, chúng ta có thể sử dụng công cụ phát triển GPDK (Grid Poratl Development Kits) [30].

1.4.5.1 Các yêu cầu đối với Grid Portal

1. Các yêu cầu về an toàn bảo mật: hiện tại, khi phát triển các Grid Portal, một số yêu cầu sau liên quan tới vấn đề bảo mật như quản lý giấy ủy nhiệm lưới, giao tiếp với cơ sở hạ tầng bảo mật GSI thông qua Portal:

- Người dùng phải có một trình duyệt Web chuẩn để tiếp cận các Grid Portal
- Người dùng có thể truy nhập tại những điểm mà họ không có sẵn giấy chứng nhận Grid
- Người dùng có thể làm bất cứ điều gì thông qua Grid Portal trong phạm vi cho phép của giấy ủy nhiệm

2. Quản lý các file từ xa: người dùng có thể truy cập các dịch vụ thư mục và file ở xa và có thể sử dụng các công cụ phân loại, lặp các bản sao dữ liệu.

3. Quản lý công việc thực hiện từ xa: người dùng có thể thực thi và quan sát các công việc thực hiện từ xa, đồng thời có thể thực thi các công việc phức tạp theo.

4. Truy cập các dịch vụ thông tin lưới: người dùng có thể tìm kiếm, truy vấn các yêu cầu thông qua các dịch vụ lưu trữ thông tin của toàn bộ hệ thống lưới.

5. Giao diện ứng dụng: cho phép ẩn đi nền tảng phức tạp của cơ sở hạ tầng lưới. Người dùng tương tác với lưới thông qua một giao diện thân thiện.

1.4.5.2 Chuyển tải các Job trong Grid Portal

Một hạ tầng bảo mật GSI mở rộng từ phiên bản của SSH có thể tương tác với các jobs được tải lên hệ thống Grid. Tất cả các jobs được tải lên sẽ được bộ quản trị GRAM gửi đến các Gatekeepers.

Các thành phần chính của công cụ GDPK dùng để chuyển tải các job là JobBean, JobSubmissionBean và JobInfoBean. Bên cạnh ba thành phần chính, GDPK còn cung cấp hai thành phần bean phụ khác có liên quan đến job là JobMonitorBean và JobHistoryBean.

- ***Truyền tập tin (File Tranfer)***

Việc di chuyển tập tin hiện nay được cung cấp bởi thành phần FileTranferBean. Nó có hai dạng:

- *GSISCPTranferBean*: được sử dụng đối với các hệ thống có cài đặt GSISSH giúp di chuyển file thông qua cơ chế GSI được mở rộng từ "scp"
- *GSIFPTTranferBean*: cài đặt GSI mở rộng từ FTP thuần java và nó có khả năng di chuyển file giữa các third-party khác nhau

Cả hai dạng trên đều bảo đảm an toàn cho việc sao lưu dữ liệu từ máy nguồn đến máy đích thông qua cơ chế xác thực user.

- ***Hồ sơ người dùng (User Profile)***

User profile hoạt động dựa trên ba thành phần bean sau :

- *UserLoginBean*: cung cấp một dịch vụ lựa chọn để xác thực user trong quá trình login vào Portal. Nó chỉ khởi tạo username, password dựa vào các file mật mã lưu trữ tương ứng trên web server để tạo ra một truy cập hợp pháp
- *UserAdminBean*: cung cấp khả năng sắp xếp việc xử lý tuần tự cho UserProfileBean và hợp thức hóa các user profile được bảo vệ bằng password
- *UserProfileBean*: dùng để duy trì các thông tin trước đây về user như thông tin xác thực, tài nguyên tính toán, các job đã submit trong quá khứ … UserProfileBean được khởi tạo cùng lúc với session để theo dõi trong suốt quá trình user thao tác trên portal

1.4.6 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,...

1.4.6.1 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.

1.4.6.2 Yêu cầu đối với một hệ thống giám sát lưới

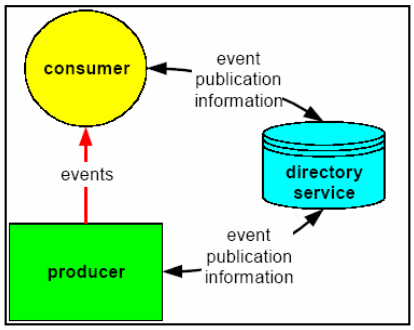
Một hệ thống giám sát lưới phải thỏa mãn được những yêu cầu sau đây:

- *Khả năng mở rộng*: phải hoạt động tốt khi số lượng tài nguyên và người dùng tăng lên
- *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 thành phần của 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

1.4.6.3 Kiến trúc bộ giám sát lưới GMA (Grid Monitoring Architecture)

Tổ chức lưới toàn cầu GGF (Global Grid Forum), đã đề xuất một kiến trúc chung cho các hệ thống giám sát lưới gọi là kiến trúc GMA. Ý tưởng của GMA là tách biệt thao tác tìm kiếm dữ liệu với thao tác truyền dữ liệu và xây dựng các thành phần có khả năng hoạt động trong theo cấu trúc phân cấp.



Hình 1-11 Các thành phần của kiến trúc GMA

1. Dịch vụ thư mục (Directory Service): phục vụ việc mô tả và khám phá các sự kiện. Nó lưu trữ thông tin về các bộ sinh và bộ tiêu thụ có khả năng tham gia tương tác. Bốn chức năng của một dịch vụ thư mục là: *Thêm, Cập nhật, Xóa* một mục thông từ thư mục; *Tìm kiếm* các bộ sinh hay bộ tiêu thụ dựa trên một số tiêu chí.

2. Bộ sinh (Producer): là bất kì thành phần nào sử dụng giao diện sinh để gửi các sự kiện tới một bộ tiêu thụ. Một bộ sinh cung cấp các chức năng cơ bản như: *Bảo trì bản đăng kí*; *Chấp nhận truy vấn*; *Chấp nhận đăng kí*; *Chấp nhận ngừng đăng kí*; *Định vị bộ tiêu thụ*; *Thông báo*; *Khởi tạo đăng kí*; *Khởi tạo ngừng đăng kí*

3. Bộ tiêu thụ (Consumer): bộ tiêu thụ là bất kì thành phần nào cài đặt giao diện tiêu thụ để nhận các sự kiện từ một bộ sinh. Một bộ tiêu thụ có các chức năng cơ bản như: *Định vị bộ sinh*; *Khởi tạo truy vấn*; *Khởi tạo đăng kí*; *Khởi tạo ngừng đăng kí*; *Bảo trì bản đăng kí*; *Chấp nhận thông báo*; *Chấp nhận đăng kí*; *Chấp nhận ngừng đăng kí; Định vị sơ đồ sự kiện*.

- ***Các tương tác với dịch vụ thư mục***

Các bộ sinh công bố sự tồn tại của mình nhờ các mục thông tin trong dịch vụ thư mục. Nhờ đó, dịch vụ thư mục được dùng để định vị các bộ sinh và bộ tiêu thụ. Các bộ tiêu thụ sử dụng dịch vụ thư mục để tìm kiếm các bộ sinh phù hợp với yêu cầu của mình, ngược lại các bộ sinh cũng dùng dịch vụ thư mục để tìm kiếm các bộ tiêu thụ phù hợp. Cả bộ sinh lẫn bộ tiêu thụ đều có thể đóng vai trò khởi tạo tương tác khi đã tìm được đối tác phù hợp, khi đó các thông báo điều khiển và các dữ liệu giám sát sẽ được truyền trực tiếp giữa nơi sinh và nơi nhận mà không liên quan đến dịch vụ thư mục nữa.

- ***Các tương tác giữa Bộ sinh và Bộ tiêu thụ***

Kiến trúc GMA hỗ trợ ba kiểu giao tiếp giữa bộ sinh và bộ tiêu thụ. Đó là:

1. Xuất bản/đăng kí: kiểu giao tiếp này gồm ba giai đoạn. Giai đoạn đầu tiên, nút khởi tạo (bộ sinh hoặc bộ tiêu thụ) liên hệ với một nút phục vụ và thông báo các tập sự kiện mà nó quan tâm. Ở giai đoạn hai, bộ sinh (nút khởi tạo hoặc nút phục vụ) gửi các sự kiện cho bộ tiêu thụ. Giai đoạn cuối cùng, bộ sinh hoặc bộ tiêu thụ chấm dứt tương tác bằng các thông báo điều khiển.

2. Truy vấn/trả lời: trong kiểu tương tác này, nút khởi tạo bắt buộc phải là một bộ tiêu thụ. Giai đoạn một là giai đoạn thiết lập tương tự như giai đoạn một của tương tác xuất bản/đăng kí. Tuy nhiên trong giai đoạn hai, tất cả các sự kiện được truyền một lần duy nhất tới bộ tiêu thụ mà không có thông báo kết thúc.

3. Thông báo: là kiểu tương tác một giai đoạn, nút khởi tạo là một bộ sinh. Bộ sinh sẽ truyền tất cả các sự kiện tới bộ tiêu thụ trong một thông báo duy nhất.

1.4.6.4 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 kiên 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 tiểu biểu thuộc loại này là Globus MDS.

1.5 Kết chương

Chương này tìm hiểu được các khái niệm, cấu trúc, thành phần của một lưới tính toán. Đồng thời cũng giới thiệu về các chuẩn kiến trúc được dùng trong tính toán lưới và các dịch vụ chính cần cung cấp trong một lưới…Qua đó ta thấy rằng, để xây dựng được một lưới tính toán chỉ dựa vào hạ tầng phần cứng là không thể, điểm quan trọng trong lưới là cơ sở hạ tầng phần mềm. Hiện tại, đã có nhiều bộ công cụ phần mềm hỗ trợ cho việc triển khai một lưới, trong đó bộ công cụ nổi tiếng nhất và được sử dụng rộng rãi nhất hiện nay là Globus Toolkit. Chi tiết về bộ công cụ này sẽ được trình bày ở Chương 2 của luận văn.

CHƯƠNG 2. TỔNG QUAN VỀ GLOBUS

Globus là một hạ tầng cung cấp các công cụ phần mềm để xây dựng tính toán mạng lưới và các ứng dụng khác dựa trên công nghệ Grid một cách dễ dàng. Các công cụ này được gọi chung là Globus Toolkit (GT). 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.

Chương này giới thiệu chủ yếu là về phiên bản 4 của GT (GT4), nó chính thức được ban hành tháng tư năm 2005 với nhiều chức năng mới của dịch vụ Web như: các thành phần, tính tương thích với các chuẩn, tính tiện lợi, và chất lượng tài liệu.

2.1 Tổng quan kiến trúc chung của GT

2.1.1 Các chức năng chính của GT

Vì GT được coi là công cụ nền dùng để xây dựng lưới tính toán, vì vậy ta sẽ nghiên cứu kiến trúc của GT bằng cách ánh xạ tương ứng các thành phần chức năng chính của lưới sang các thành phần chức năng chính của GT [6] bao gồm:

- *Định vị và cấp phát tài nguyên*: cung cấp kỹ thuật dùng cho các ứng dụng đòi hỏi tài nguyên ngay lập tức hoặc tài nguyên được đặt trước. Đây là chức năng bắt buộc phải có vì ứng dụng không thể xác định đúng vị trí của tài nguyên cần thiết nhất là khi các tài nguyên có sẵn bị thay đổi
- *Truyền thông*: cung cấp kỹ thuật truyền thông cơ bản, cho phép thực thi hiệu quả việc truyền đạt thông tin trên diện rộng bao gồm việc truyền thông điệp, gửi thủ tục, phân phối bộ nhớ…với độ tin cậy cao trên một băng thông rộng
- *Thống nhất thông tin phục vụ*: cung cấp kỹ thuật đồng bộ thông tin về cấu trúc và trạng thái thực của hệ thống metasystem. Ngoài ra, nó phải có thành phần cung cấp thông tin và hỗ trợ việc điều khiển yêu cầu, truy cập thông tin
- *Giao diện xác thực*: cung cấp những kỹ thuật cơ bản về xác thực dùng để xác nhận tính hợp lệ của người dùng và tài nguyên

- *Tạo tiến trình hoạt động*: được sử dụng để thực hiện tính toán trên tài nguyên đã được định vị và phân phát. Nó bao gồm: thực thi sự điều chỉnh, tạo ra môi trường thực thi, chuyển đổi số, kết hợp với tiến trình mới vào, quản lý sự kết thúc và tiến trình shutdown
- *Truy xuất dữ liệu*: chịu trách nhiệm truy xuất dữ liệu ở tốc độ cao để lưu trữ dưới dạng các tập tin

2.1.2 Các đặc trưng của GT4

➢ ***Các cơ chế và giao diện được chuẩn hoá***

Trong môi trường tính toán lưới, các thành phần thường tương tác với nhau qua mạng. Do cách xây dựng hệ thống phân tán là không đồng nhất nên cần thiết phải thự hiện chuẩn hoá các cơ chế xử lý công việc chung. Việc chuẩn hoá trong môi trường lưới nhằm mục đích:

- Dễ dàng xây dựng và hiểu các thành phần riêng lẻ
- Dễ liên thông giữa các cài đặt khác nhau của cùng một giao diện
- Dễ chia sẻ các thành phần và phát triển các công cụ cho phép dùng lại

GT4 chọn cơ chế dịch vụ Web để định nghĩa giao diện và cấu trúc các thành phần [6]. Các dịch vụ Web này cung cấp cơ chế mềm dẻo, mở dựa trên XML để mô tả, phát hiện, và thực thi các dịch vụ từ xa. Giao diện dịch vụ Web được GT4 dùng cho hầu hết các thành phần chính, do đó cho phép sử dụng các cơ chế dịch vụ Web chuẩn để mô tả các giao diện dịch vụ GT4.

Trong một vài năm trở lại đây, nhiều phần mềm đã được phát triển để hỗ trợ triển khai nhiều thành phần hệ thống phân tán có cài đặt giao diện dịch vụ Web. GT4 dùng GT4 Web Services Containers để triển khai và quản lý các dịch vụ GT4 được viết bằng nhiều ngôn ngữ như C, Java. Tuy nhiên, ta cần chú ý là dịch vụ Web thường không chú trọng vào hiệu năng của hệ thống.

➢ ***Cơ sở hạ tầng***

Ứng dụng của người dùng thường yêu cầu tầng trung gian của GT4 phải cung cấp các công cụ duy trì và quản lý cơ sở hạ tầng phần cứng của Grid. Các khái niệm hướng dịch vụ và các cơ chế dịch vụ Web được dùng để truy xuất và quản lý

các phần tử hạ tầng thông qua các giao diện của chúng. GT4 cài đặt các giao diện dịch vụ Web để: quản lý các tài nguyên tính toán và theo dõi quá trình hoạt động của các tài nguyên này (GRAM); quản lý thiết bị đo đạc (GTCP); và quản lý việc truyền dữ liệu tin cậy (RFT). Ngoài ra, GT4 còn cung cấp dịch vụ GridFTP nhưg chưa được định nghĩa giao diện dịch vụ Web.

- ***Phát hiện và theo dõi***

Phát hiện và theo dõi là 2 chức năng quan trọng trong lưới. Việc phát hiện tài nguyên hoặc dịch vụ phù hợp với yêu cầu là cần thiết, trong khi việc theo dõi giúp duy trì hoạt động của các tài nguyên hoặc dịch vụ trong hệ thống.

Đầu tiên, GT4 cung cấp các cơ chế chuẩn để gắn kết các thuộc tính tài nguyên dựa trên XML với thực thể mạng và truy xuất các thuộc tính này bằng cơ chế kéo-đẩy. Các cơ chế này là những cài đặt của đặc tả WSRF và WSNotification được xây dựng trong mọi dịch vụ và container của GT4. Thứ hai, GT4 cung cấp ba dịch vụ kết [6] nhằm thu thập thuộc tính trạng thái từ các nguồn thông tin đã đăng ký. Ba dịch vụ này cài đặt một bản đăng ký (MDS-Index), một bộ lưu trữ (MDS-Archive), và một bộ lọc dữ liệu hướng sự kiện (MDSTrigger). Cuối cùng, đối với chức năng phát hiện và theo dõi, GT4 còn cung cấp các giao diện trình duyệt, các lệnh, và các giao diện dịch vụ Web để người dùng truy vấn và nhận thông tin.

- ***Bảo mật***

Bảo mật là yếu tố rất quan trọng trong bất cứ hệ thống nào, nhất là đối với hệ thống phân tán gồm nhiều tài nguyên và người sử dụng nằm rải rác nhiều nơi như Grid. GT4 và các công cụ liên quan cung cấp nền tảng bảo mật hoàn chỉnh không chỉ đối với truyền thông trên mạng mà còn đối với các tài nguyên và người dùng.

- ***Dữ liệu***

Trong lưới, dữ liệu thường rất lớn và nằm rải rác. Vì vậy chúng cần được quản lý, tích hợp lại để truy xuất nhanh chóng. GT4 có rất nhiều thành phần hỗ trợ vấn đề này như GridFTP, RLS, DAI (cung cấp khả năng truy xuất tới dữ liệu XML và các cơ sở dữ liệu quan hệ) [6].

- ***Phối hợp***

GT4 có nhiều công cụ phối hợp mức cao như DAGman và MPICH-G2.

2.1.3 Tóm lược về kiến trúc của GT4

Tóm lược kiến trúc của GT4 [6] gồm:

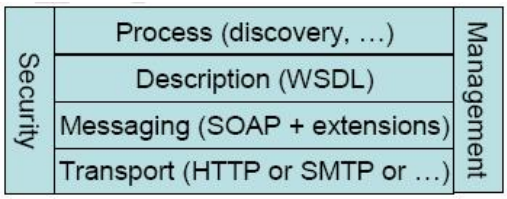
- *Kiến trúc hướng dịch vụ*: gồm nhiều dịch vụ hoàn chỉnh (đi kèm các thư viện chuẩn) tương tác với nhau qua các giao thức chuẩn hoá
- *Dịch vụ hạ tầng*: gồm các dịch vụ truy xuất, theo dõi, quản lý và điều khiển các phần tử hạ tầng
- *Dịch vụ Web*: tuân theo các giao thức dịch vụ Web đã được sử dụng rộng rãi và các cơ chế để mô tả, phát hiện, truy xuất, xác thực, phân quyền dịch vụ…
- *GT4 Containers*: gồm các thành phần để tạo nên GT4 container – nơi chứa các dịch vụ Web viết bằng Java, C và Python
- *Bảo mật*: cho phép thiết lập một hệ thống bảo mật cao, mở, và uyển chuyển nhằm bảo vệ thông tin, xác thực và phân quyền người dùng, đại diện và ủy quyền
- *Chuẩn hoá*: được cài đặt theo các chuẩn đã được chấp nhận rộng rãi, hỗ trợ tạo các thành phần mở, dễ dàng sử dụng lại nhờ các công cụ chuẩn
- *Công cụ liên quan*: GT4 là tầng trung gian cung cấp nền tảng tốt nhất cho các ứng dụng lưới
- *Hạn chế*: GT4 còn nhiều hạn chế về hiệu năng, tính dễ sử dụng, và sức mạnh

2.2 Kiến trúc hướng dịch vụ

Phần này sẽ nói đến các khái niệm cơ bản về dịch vụ Web và việc sử dụng cũng như cài đặt chúng trong GT4, nhất là trong các thành phần lõi của GT.

2.2.1 GT4, các hệ thống phân tán, các dịch vụ Web

GT4 gồm các phần mềm cài đặt cơ chế dịch vụ Web để xây dựng các hệ thống phân tán, các thống này luôn phải đối đầu với các vấn đề như độ trễ mạng, sự cố cục bộ, tương tranh. Ngoài ra, các thành phần hệ thống có thể nằm rải rác trên các vùng địa lý khác nhau và được quản trị riêng biệt.



Hình 2-1 Các thành phần trong kiến trúc dịch vụ Web

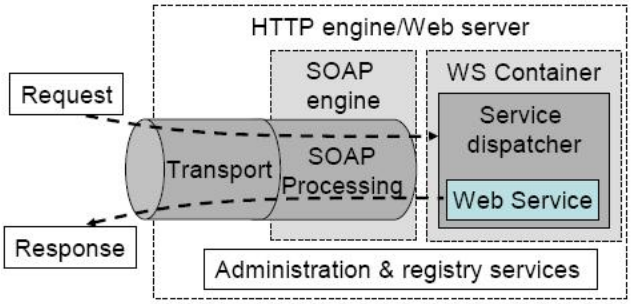
Một dịch vụ Web [27] là một hệ thống phần mềm được thiết kế để hỗ trợ tương tác giữa máy với máy (*machine-to-machine*) trên mạng, nói cách khác cho phép các phần mềm dễ dàng tương tác với nhau theo một chuẩn thống nhất. Mỗi dịch vụ có một giao diện được mô tả theo khuôn dạng, ngôn ngữ chuẩn (WSDL) giúp người lập trình và các phần mềm hoặc dịch vụ khác dễ dàng phân tích và hiểu được dịch vụ đó cung cấp cái gì, như thế nào. Thông điệp được dùng để trao đổi giữa các dịch vụ tuân theo giao thức SOAP.

2.2.2 Cơ sở hạ tầng và ứng dụng hướng dịch vụ

Các công nghệ dịch vụ Web-GT4 có thể dùng để xây dựng cả cơ sở hạ tầng hướng dịch vụ và ứng dụng hướng dịch vụ. Các ứng dụng hướng dịch vụ gồm nhiều dịch vụ giao dịch với nhau. Cơ sở hạ tầng hướng dịch vụ gồm các dịch vụ nền tảng cung cấp các cơ chế quản lý tài nguyên phần cứng và phần mềm phân tán.

- ***Cài đặt dịch vụ Web***

Để xây dựng một ứng dụng hướng dịch vụ Web, ta cần phải cài đặt 2 phần là: *Môi trường chủ (container)*: đảm nhiệm các chức năng trao đổi với client bằng thông điệp SOAP, xác định và gọi dịch vụ để xử lý thông điệp, quản lý, điều hành các dịch vụ Web. *Cài đặt dịch vụ Web*: viết đoạn mã để xử lý thông điệp gửi tới.



Hình 2-2 Các thành phần chức năng chính trong cài đặt dịch vụ Web

- ***Đặc tả dịch vụ Web***

Đây là các đặc tả dịch vụ Web đã được cài đặt trong GT4 [6]:

XML được dùng trong dịch vụ Web để định dạng dữ liệu chuẩn, mềm dẻo và mở rộng. GT4 cho phép XML xuất dữ liệu dạng text và nhị phân.

SOAP 1.2 cung cấp cơ chế chuẩn, mở rộng và hướng thành phần để đóng gói và trao đổi thông điệp XML giữa nhà cung cấp dịch vụ và khách hàng. SOAP độc lập với nghi thức truyền thông, tuy nhiên thông thường HTTP được dùng như giao thức nền tảng.

WSDL 1.1 là một tài liệu XML để mô tả dịch vụ Web, bao gồm các qui ước gắn kết chuẩn giữa SOAP với các giao thức nền khác.

WS-Addressing định nghĩa cơ chế đánh địa chỉ cho dịch vụ web và thông điệp. Nó định nghĩa các phần tử XML để xác định các endpoint của dịch vụ Web và để đảm bảo phân biệt các endpoint này trong thông điệp.

WS Resource Framework (WSRF) định nghĩa cơ cấu mở và tổng quát cho việc mô hình hóa và truy xuất các tài nguyên.

WS-Notification định nghĩa nền cho phép phổ biến thông tin giữa các dịch vụ Web.

Ngoài ra GT4 còn bao gồm một số cài đặt mở rộng khác như: **GridFTP** là mở rộng của FTP; **DAIS** tuân theo đặc tả OGSA-DAIS của diễn đàn lưới toàn cầu [6].

2.2.3 Kiến trúc hướng dịch vụ (Service Oriented Architecture-SOA)

Kiến trúc hướng dịch vụ SOA là một form của kiến trúc các hệ thống phân tán có các đặc trưng sau [6]:

1. Logical view: dịch vụ là một cách nhìn logic, trừu tượng của chương trình, cơ sở dữ liệu, quy trình nghiệp vụ…

2. Message orientation: dịch vụ được định nghĩa theo thông điệp trao đổi giữa người cung cấp và người yêu cầu chứ không phải về bản thân các đối tác. Cấu trúc bên trong của đối tác được trừu tượng hoá trong kiến trúc SOA.

3. Description orientation: dịch vụ được mô tả như siêu dữ liệu nhưng dễ dàng được xử lý bằng máy tính. Cách thức sử dụng dịch vụ và ngữ nghĩa của nó là nội dung chính cần mô tả.

4. Granularity: dịch vụ chỉ gồm một số ít các thao tác ứng với các thông điệp dài và phức tạp.

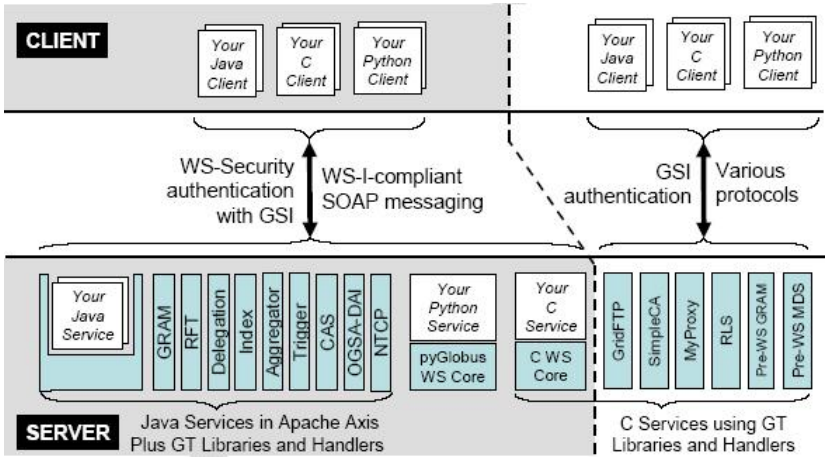
5. Network orientation: dịch vụ có xu hướng sử dụng mạng máy tính.

6. Platform neutral: thông điệp được trao đổi trên một nền tảng trung lập, được chuẩn hoá khuôn dạng và được phân phối thông qua các giao diện. XML là lựa chọn tốt nhất để thoả mãn điều kiện này.

2.3 Kiến trúc GT4

2.3.1 Kiến trúc tổng quan

Hình sau mô tả kiến trúc của GT4 bao gồm các dịch vụ được cài đặt bên phía chủ và các thư viện gắn với phía khách. GT4 cung cấp các thành phần dịch vụ Web (bên trái) và cả thành phần không là dịch vụ Web (bên phải) [6].



Hình 2-3 Các thành phần trong kiến trúc GT4

Các hình hộp trắng trong phần "Client" chỉ các ứng dụng hoặc công cụ truy xuất đến dịch vụ GT4 hoặc các dịch vụ được phát triển trên GT4.

- ***Các dịch vụ GT4 được định nghĩa trước***

GT4 cung cấp các dịch vụ Web đã có giao diện, đó là: GRAM, RFT, ủy quyền, dịch vụ theo dõi và phát hiện tài nguyên (MDS-Index, MDS-Trigger), MDSArchive, dịch vụ phân quyền trong cộng đồng, dịch vụ tích hợp và truy xuất dữ liệu (OGSA-DAI) và dịch vụ điều khiển thiết bị từ xa (GTCP - Grid TeleControl Protocol) [6].

GRAM và MDS-Index tuy đã lỗi thời nhưng vẫn được cung cấp trong GT4. Ba dịch vụ GridFTP, RLS và MyProxy (kho uỷ nhiệm thư trực tuyến) tuy được cài đặt nhưng chưa có giao diện dịch vụ Web đầy đủ. Ngoài ra GT4 còn cung cấp các thư viện để liên kết với các cơ chế xác thực và ủy quyền khác như MD5, LDAP, Kerberos, cũng như thư viện IO mở rộng (XIO) cho phép truy xuất nhiều giao thức vận chuyển cơ sở. Ta có thể sử dụng SimpleCA như là một cơ chế cấp chứng thư gọn nhẹ.

- ***GT4 Containers***

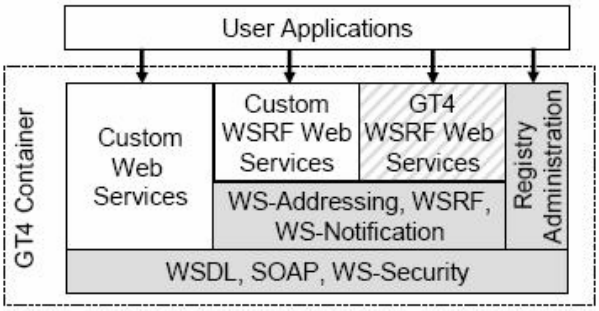
GT4 đã cài đặt các dịch vụ Web quan trọng cho ứng dụng lưới, một số trong chúng đã được chuẩn hoá như WSRF, WS-Notification, và một số chỉ có trong Globus như GRAM, RFT. Các dịch vụ này có thể kết hợp với các thành phần khác (máy phục vụ Web, máy SOAP…) để tạo ra nhiều loại GT4 container khác nhau.

Một GT4 container [6] là một "thùng đựng" chứa các dịch vụ Web với các đặc trưng sau:

- Cài đặt giao thức trao đổi thông điệp SOAP trên nền HTTP cũng như cơ chế bảo mật mức giao vận và mức thông điệp WS-Sercurity cho quá trình truyền thông
- Cài đặt các dịch vụ WS-Addressing, WSRF, và WS-Notification
- Hỗ trợ ghi chép qua Log4j mà cài đặt giao diện lập trình Jakarta Commons Logging API
- Định nghĩa tài nguyên dịch vụ Web (WS-Resources) với các thuộc tính cung cấp thông tin về các dịch vụ được triển khai trong container và các thuộc tính của container như phiên bản và thời điểm khởi động

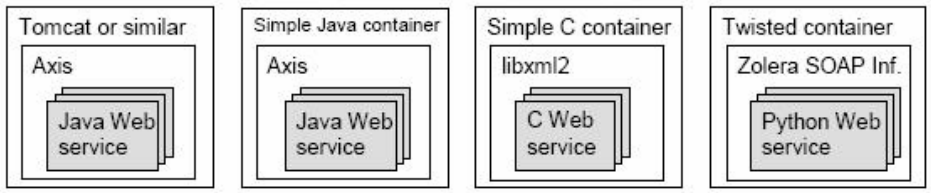
vì thế một GT4 container có thể chứa các dịch vụ mà giao diện của chúng được định nghĩa theo đặc tả dịch vụ Web cơ bản và các dịch vụ mà giao diện của chúng theo nền WSRF. Ngoài ra GT4 Java container còn có thể chứa các dịch vụ nâng cấp cung cấp bởi Globus như GRAM, MDS, và RFT.

Các ứng dụng phía client có thể sử dụng giao diện đã đăng ký trong GT4 container đến xác định dịch vụ nào hiện hữu trong container, và truy xuất giao diện quản trị GT4 container để thực hiện các chức năng quản trị cơ bản.



Hình 2-4 GT4 Container tích hợp các dịch vụ và công cụ

GT4 hỗ trợ 3 ngôn ngữ để cài đặt dịch vụ Web là Java, C, và Python:



Hình 2-5 Bốn cấu hình GT4 container

2.3.2 Triển khai dịch vụ Web trên GT4

Triển khai một dịch vụ Web bao gồm các công việc cài đặt và khởi tạo việc thực thi một dịch vụ Web trên một máy tính cụ thể hay nói rõ hơn trên một container. Bởi vì một container dịch vụ Web có thể được cấu hình cho một mục đích cụ thể, nên việc triển khai dịch vụ Web có thể bao hàm:

1. Triển khai dịch vụ Web vào trong một GT4 container.
2. Tải các thư viện bổ sung (WSRF) vào một non-GT4 container, sau đó triển khai dịch vụ Web vào container này.

3. Triển khai một GT4 container mới, sau đó mới triển khai dịch vụ Web hoặc
4. Triển khai GT4 container và dịch vụ Web đồng thời.

Gói phần mềm GT4 cung cấp các công cụ để dễ dàng thực hiện công việc này trong các môi trường khác nhau. Chẳng hạn như, triển khai dịch vụ GT4 GRAM thì cần:

1. Nếu GT4 Java container đã được cài đặt, ta có thể triển khai dịch vụ GRAM một cách trực tiếp.
2. Nếu máy phục vụ đã cài đặt Tomcat, nhưng chưa cài gói Globus nào thì ta cần triển khai các dịch vụ WSRF, WS-Notification, và sau đó là GRAM.
3. Nếu chưa có gì thì ta nên cài đặt gói GT4 hoàn chỉnh gồm GT4 Java container và dịch vụ Web GRAM. Gói này có thể cài đặt, cấu hình, và chạy dễ dàng.

2.4 Quản lý thực thi trong GT4

Các công việc liên quan tới quản lý thực thi trong môi trường phân tán [6] thường là: tạo một chương trình chạy như một dịch vụ mạng; phân bổ; chạy một chương trình thực thi từ xa; chạy một chương trình song song trên các máy tính phân tán; và chạy một tập các công việc không thường xuyên trao đổi dữ liệu với nhau.

Các công việc này đòi hỏi phải cần đền các công cụ quản lý thực thi như: khởi tạo; theo dõi; quản lý; lập lịch và điều phối các công việc tính toán từ xa trong môi trường phân tán.

GT4 cung cấp gói "Quản lý và cấp phát tài nguyên lưới - GRAM" để quản lý thực thi. Thông thường GRAM được triển khai cùng với các gói MyProxy và RFT nhằm tăng tính bảo mật, xác thực và trao đổi dữ liệu.

Gói GRAM gồm 3 thành phần chính: *Bộ lập lịch GRAM* cho các cụm máy trong một mạng cục bộ (Condor, OpenPBS, Torque, PBSPro, SGE, LSF). *Hệ thống cung cấp các giao diện cho các máy tính ở xa* (OpenSSH) hoặc cài đặt các mô hình lập trình song song khác nhau trong môi trường lưới bằng cách dùng GRAM phân bổ các công việc tính toán tới các máy tính từ xa (Condor-G, DAGman, MPICH-

G2, GriPhyN VDS, Nimrod-G). Cuối cùng là *Bộ siêu lập lịch* để ánh xạ công việc tới các nhóm (CSF, Maui).

2.4.1 Tổng quan về GT4 GRAM

Để người dùng ở xa thực thi một chương trình thông qua một dịch vụ Web, ta cần phải định nghĩa và cài đặt một dịch vụ Web gồm một phương thức gọi thực thi chương trình từ xa. Tuy nhiên, để cài đặt được ta phải giải quyết các vấn đề:

1. State: công việc tính toán có thể thực hiện các thao tác nhập/xuất trong khi chạy làm ảnh hưởng tới trạng thái của tài nguyên tính toán hoặc hệ thống tập tin gắn với công việc này.

2. User executables: người dùng có thể cung cấp chương trình của riêng họ để gửi thực thi từ xa.

3. Staging of input and output: chương trình thực thi, dữ liệu vào/ra có thể lớn, ở xa, và/hoặc được chia sẻ với các lời gọi thực thi khác. Vì thế, yêu cầu bố trí dữ liệu vào/ra là cần thiết.

4. Streaming output: một số chương trình thực thi cần có khả năng cung cấp kết quả kịp thời cho người dùng khi đang chạy.

5. Control: người dùng đôi khi cần ngưng một công việc đang thực thi.

6. Scheduler: các tài nguyên tính toán lớn thường chịu sự điều khiển của một bộ lập lịch để cấp phát tài nguyên theo các chính sách ưu tiên một cách tối ưu về hiệu năng.

7. Monitoring: đối với một số công việc phức tạp cần được theo dõi quá trình thực thi và các thao tác can thiệp kịp thời như pending, suspending, staging, …

Hệ thống GRAM kết hợp với các công cụ quản lý đã có được phát triển nhằm giải quyết các yêu cầu trên. GRAM cho phép: thực thi một chương trình bất kỳ; hoàn thành các thao tác tin cậy; thực hiện theo dõi các trạng thái; quản lý ủy nhiệm thư; bố trí tập tin; tương tác với các bộ lập lịch đã có.

Mỗi công việc khi được đệ trình sẽ có: một thực thể stateful là ManagedJob được tạo ra trên máy chủ tính toán với chu kỳ sống ứng với công việc; một handle hay một tham chiếu địa chỉ dịch vụ Web (WS-Addressing endpoint refrence). Client

sẽ nhận được handle này và sử dụng nó để truy vấn trạng thái của công việc, hủy công việc, nhận "thông báo" khi công việc thay đổi trạng thái, và/hoặc nhận kết quả ra.

Dưới quan điểm của người quản trị hệ thống, GRAM không phải là một bộ lập lịch tài nguyên mà là một phương tiện trung gian để tương tác với các bộ lập lịch cục bộ khác thông qua một khuôn dạng thông điệp chuẩn. Gói GRAM của GT4 cho phép liên kết với các bộ lập lịch như Condor, LSF, SGE, và PBS…

2.4.2 Lệnh globusrun-ws

Lệnh này bao gồm các chức năng quản lý công việc thực thi như: gửi công việc để thực thi, theo dõi quá trình thực thi và quản lý các công việc và các chức năng tinh vi khác như: truy xuất file, ủy quyền, gửi công việc chỉ một lần, timeout, bảo mật thông điệp, và phân quyền.

- ***Chức năng Basic Job submission***

Lệnh đệ trình thực thi chương trình "/bin/touch" với tham số vào "touched_it" được gõ lệnh như sau:

```
% globusrun-ws -submit -job-command /bin/touch touched_it
```

Nếu đệ trình thành công, ta sẽ nhận được thông tin trạng thái của chương trình này khi nó đang chạy và khi kết thúc như sau (các thông tin này được hiển thị ngay dưới lệnh vừa gõ):

```
Submitting job...Done.
Job ID: uuid:c51fe35a-4fa3-11d9-9cfc-000874404099
Termination time: 12/17/2004 20:47 GMT
Current job state: Active
Current job state: CleanUp
Current job state: Done
Destroying job...Done.
```

Nếu không thích gõ lệnh, ta có thể mô tả công việc cần đệ trình trong một file XML, và lệnh gửi thực thi công việc này như sau:

```
% cat touch.xml
<job>
<executable>/bin/touch</executable>
<argument>touched_it</argument>
```

```
</job>

% globusrun-ws -submit -job-description-file touch.xml
```

Để xác định nơi chương trình sẽ thực thi, ta phải mô tả một tham chiếu tới điểm cuối trong một file.epr như:

```
% cat gram.epr
<factoryEndpoint
mlns:gram=http://www.globus.org/namespaces/2004/10/gram/job
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/03/addressing">
<wsa:Address>
https://viz-
login.isi.edu:9000/wsrf/services/ManagedJobFactoryService
</wsa:Address>
<wsa:ReferenceProperties>
<gram:ResourceID>PBS</gram:ResourceID>
</wsa:ReferenceProperties>
</factoryEndpoint>

% globusrun-ws -submit -job-description-file touch.xml -
factory-epr-file gram.epr
```

Giá trị mặc định của *-factory* và *-factory-type* lần lượt là "*localhost*" và "*fork*". Trong ví dụ trên thì công việc sẽ được thực thi trên máy hiện tại bằng lời gọi fork() của hệ điều hành UNIX.

```
% globusrun-ws -submit -factory localhost -factory-type fork
\
-job-command /bin/touch touched_it
```

- ***Tương tác với công việc đã gửi***

Có 2 cách để tương tác với công việc sau khi gửi là: tương tác *Online*, nghĩa là tạo và duy trì một liên kết liên tục với công việc đã gửi để nhận được thông tin về trạng thái công việc cũng như ra lệnh ngưng giữa chừng; tương tác *Offline hoặc Batch mode*, nghĩa là không duy trì liên kết liên tục mà trả về một tham chiếu đến điểm cuối EPR của công việc. Người dùng có thể yêu cầu cho biết trạng thái công việc khi cần. Để làm điều này, ta dùng tùy chọn *-job-epr-output-file* tiếp theo là file sẽ chứa EPR trong lệnh đệ trình công việc phía trên [6].

Để nhận được trạng thái hiện tại của công việc, ta gõ lệnh:

```
% globusrun-ws -status -job-epr-file mj.epr
```

hoặc chuyển sang chế độ online:

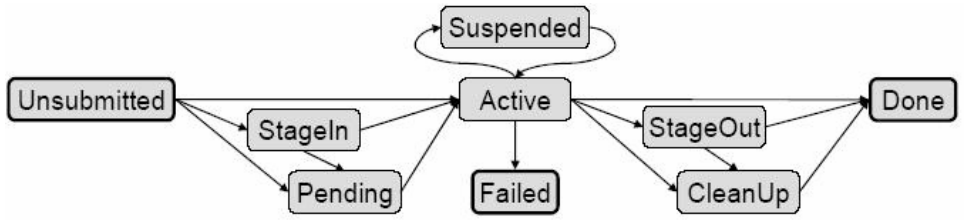
```
% globusrun-ws -monitor -job-epr-file mj.epr
```

hoặc ngưng (kill) công việc:

```
% globusrun-ws -kill -job-epr-file mj.epr
```

- ***Trạng thái và chu kỳ sống của công việc***

Hình sau mô tả sơ đồ dịch chuyển trạng thái trong chu kỳ sống của công việc.



Hình 2-6 Sự dịch chuyển trạng thái

- *Unsubmitted*. Công việc chưa được gửi
- *StageIn*. Đang đợi để truy xuất file input
- *Pending*. Bộ lập lịch cục bộ vẫn chưa lập lịch cho công việc đệ thực thi
- *Active*. Công việc đang thực thi
- *Suspended*. Tạm ngưng thực thi công việc
- *StageOut*. Công việc đã thực thi xong, đang đợi để ghi kết quả ra
- *CleanUp*. Đã thực thi xong, hoàn thành in kết quả, thực hiện dọn dẹp
- *Done*. Hoàn thành mọi việc
- *Failed*. Bị lỗi

- ***Bố trí file vào, ra, lỗi và các đặc tính khác***

Trong nhiều trường hợp, tập tin thực thi và dữ liệu vào cần được đưa đến máy đích và/hoặc dữ liệu ra phải được đưa ra. Việc bố trí tập tin này được đặc tả bằng các chỉ thị chuyển tập tin trong file mô tả công việc. Những chỉ thị này tuân theo cú pháp RFT gồm địa chỉ URL nguồn và đích [6].

- ***Đại diện tin cậy***

Một công việc gửi tới máy chủ GRAM nhiều khi cần có ủy nhiệm thư nếu muốn:

- Thực hiện các thao tác từ xa, như gửi công việc GRAM, hay gọi các dịch vụ Web khác. Trường hợp này cần có ủy nhiệm thư công việc – job credentials. Người dùng sẽ cung cấp uỷ nhiệm thư kèm với yêu cầu gửi thực hiện công việc. Ủy nhiệm sẽ hết tác dụng ngay khi công việc kết thúc
- Thực hiện các thao tác bố trí sắp xếp dữ liệu. Trường hợp này cần ủy quyền bố trí – staging credentials. Người dùng sẽ gửi ủy nhiệm thư cho dịch vụ Delegation chạy trên máy chủ GRAM, và chúng có thể được dùng cho nhiều yêu cầu thực thi công việc

GT4 cung cấp các công cụ sau để áp dụng 2 chế độ uỷ quyền này:

- Câu lệnh globus-credential-delegate cho phép tạo cả 2 loại uỷ nhiệm thư tới máy chủ GRAM và trả về một tham chiếu EPR tới ủy nhiệm mới được tạo. (client có thể dùng lệnh globus-credential-refresh để làm mới lại uỷ nhiệm thư)
- Sau khi đã có uỷ nhiệm thư bằng cách trên, clien có thể sử dụng uỷ nhiệm thư này bằng cách nhập tham chiếu EPR vào tập tin mô tả công việc trong các phần tử jobCredential, stagingCredential, và/hoặc transferCredential

Sử dụng các tuỳ chọn -job-delegate và -staging-delegate của lệnh gbusrun-ws để yêu cầu tạo ủy nhiệm thư đi kèm với một công việc. Ý nghĩa của các tuỳ chọn này như sau:

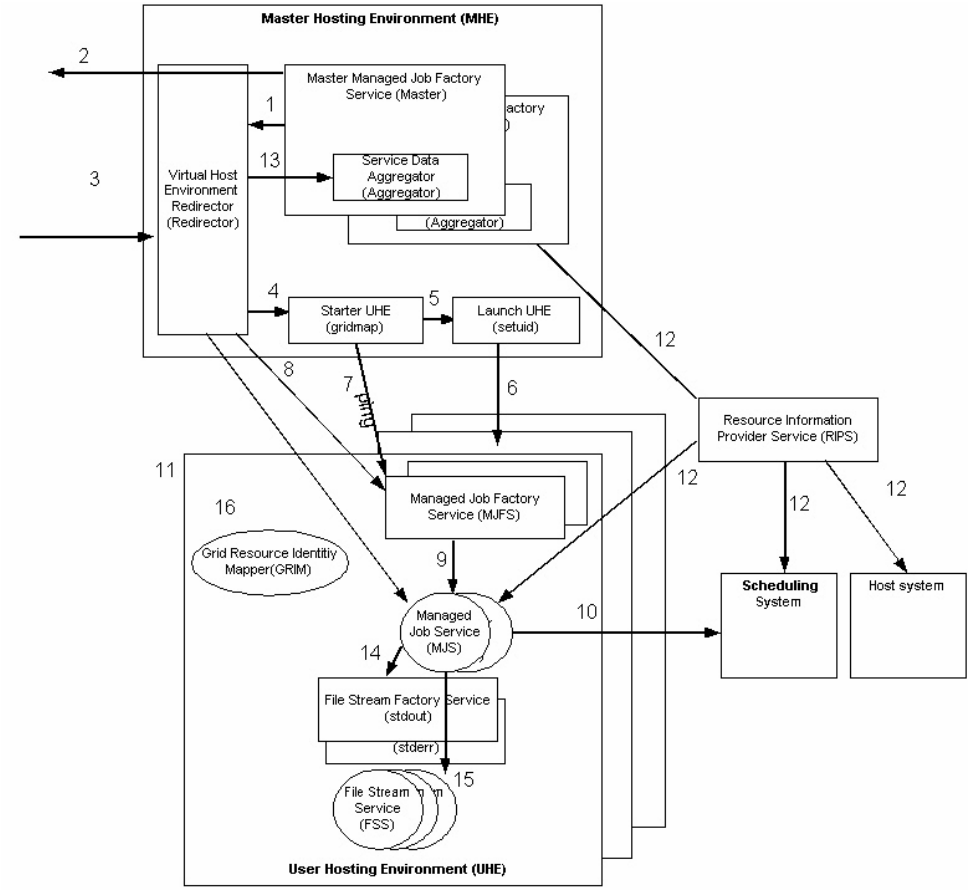
- -job-delegate: Nếu trong tập tin mô tả công việc không chứa phần tử jobCredential, globusrun-ws sẽ tạo uỷ nhiệm thư và đưa phần tử tương ứng vào file mô tả công việc
- staging-delegate: Nếu trong file mô tả công việc có chứa các chỉ thị staging hoặc cleanup và không có các phần tử stagingCredential hoặc transferCredential, globusrun-ws sẽ tự động tạo phần tử tương ứng và đưa vào trong file mô tả công việc

2.4.3 Cách thức hoạt động của GT4 GRAM

Phần này trình bày cách thức hoạt động của GRAM để hiểu rõ hơn về cơ chế hoạt động của nó cũng như cách thức phát triển dịch vụ dựa trên WSRF [15].

- ***Kiến trúc GRAM***

Kiến trúc của GRAM được mô tả như trên hình 2.7 bao gồm:



Hình 2-7 Kiến trúc GRAM

1. Virtual Host Environment Redirector: tiếp nhận các thông điệp SOAP (SOAP message) và gửi chúng đến cho môi trường người dùng .

2. Master Managed Job Factory Service – MMJFS: chịu trách nhiệm cung cấp dịch vụ chạy ứng dụng GRAM cho các yêu cầu từ bên ngoài.

3. Managed Job Factory Service – MJFS: chịu trách nhiệm tạo ra một dịch vụ quản lý công việc MJS (Managed Job Service) mới ngay khi nó nhận được một yêu cầu tạo dịch vụ. MJFS tồn tại từ khi khởi tạo môi trường người dùng (UHE) cho đến khi phiên làm việc kết thúc.

4. Managed Job Service – MJS: là một dịch vụ OGSI cho phép các đặc tả công việc có thể đệ trình công việc tới hệ thống lập lịch địa phương, theo dõi tình trạng và đưa ra các thông báo.

5. File Stream Factory Service – FSFS: được coi như là một giao diện nhận yêu cầu, chịu trách nhiệm cài đặt một dịch vụ về file (File Stream Service) mới khi nó nhận được yêu cầu tạo dịch vụ.

6. File Stream Service – FSS: một dịch vụ OGSI cho một địa chỉ URL đích sẽ đưa file địa phương vừa được tạo đến địa chỉ URL đích.

7. Resource Information Provider Service – RIPS: là một dịch vụ khai báo cung cấp thông tin về hệ thống lập lịch tài nguyên, hệ thống file, hệ thống máy chủ,

8. Grid Resource Identity Mapper – GRIM: là thành phần chịu trách nhiệm tạo ra giấy chứng nhận cho người dùng sau khi đã kiểm tra grid-mapfile.

- ***Bảo mật***

GRAM hỗ trợ các mức độ kiểm tra an toàn bảo mật:

- Cơ chế WS-Sercurity nhằm xác nhận những uỷ quyền gắn với yêu cầu và cũng để xác thực người yêu cầu.
- Phân quyền được thực hiện bằng một callout cấp quyền. Callout này sẽ truy vấn file gridmap, SAML server, hay một cơ chế cấp quyền khác.
- Nếu có đủ quyền, công việc sẽ được thực thi và trả về một số hiệu ID cục bộ. Lệnh tiện ích sudo sẽ được dùng để quản lý tài nguyên cục bộ.

Các dịch vụ chạy trong GT4 container không yêu cầu một quyền đặc biệt: các thao tác cần quyền cao sẽ được thực hiện qua hàm "sudo".

- ***Thao tác liên quan tới dữ liệu***

Các thao tác liên quan tới việc bố trí dữ liệu vào/ra của công việc được GRAM giao cho dịch vụ RFT đảm nhiệm. Tuỳ theo yêu cầu, dịch vụ RFT khởi tạo

kết nối GridFTP giữa máy nguồn và đích. Ngoài các thao tác bố trí dữ liệu chuẩn, GRAM còn hỗ trợ cơ chế cập nhật dữ liệu ra một cách liên tục khi công việc đang tiến hành.

- ***Giao diện lập trình GRAM client***

4 PortType của giao diện lập trình cho GRAM client định nghĩa các thao tác trên 4 WS-Resources mà đại diện cho trạng thái của GRAM server, trạng thái của một công việc đơn lẻ, chuỗi uỷ nhiệm thư gắn với delegation factory, và một uỷ nhiệm thư. GT4 cung cấp các gắn kết (binding) Java, C, và Python với 4 portType này:

- *PortType ManagedJobFactory*: cung cấp thao tác truy vấn thông tin về trạng thái của GRAM server. Ngoài ra, nó còn có thao tác createManagedJob để đệ trình công việc. Nếu thành công, nó sẽ trả về một WS-Resource của ManagedJob và một tham chiếu EPR của ManagedJob mới này
- *PortType ManagedJob*: định nghĩa thao tác trên WS-Resource ManagedJob cho phép truy xuất thông tin về trạng thái công việc, quản lý thời gian sống của công việc qua WS-Resource WS-ResourceLifetime, và cho phép client yêu cầu được thông báo khi có thay đổi qua WS-Resource WS-BaseNotification
- *PortType DelegationFactory*: hỗ trợ các thao tác nhằm truy vấn thông tin về các dãy uỷ nhiệm thư gắn với delegation factory. Nó còn định nghĩa một thao tác RequestSecurityToken nhằm thực hiện tải một thẻ bảo mật lên dịch vụ đại diện và trả về một tham chiếu EPR của WS-Resource mới này
- *PortType Delegation*: cung cấp các thao tác WS-ResourceLifeTime đối với thẻ bảo mật. Thao tác refresh của portType này cho phép client tải lên một uỷ nhiệm thư mới

2.4.4 Cấu hình và quản trị GT4 GRAM

Ta cần phải cài đặt một GRAM server cho phép các dịch vụ Web có thể truy xuất các tài nguyên tính toán. Hai bước chính để cấu hình gồm:

1. Local scheduler interface. GRAM có thể dùng các cơ chế, công cụ đã có sẵn để khởi tạo và điều khiển các công việc. Mặc định GRAM dùng lời gọi fork() của hệ thống. Các công cụ lập lịch theo lô như Condor, SGE, PBS, và LSF phải được cấu hình để cho phép GRAM gửi các công việc và theo dõi trạng thái hoạt động của công việc.

2. Authorization to user mapping. GRAM dùng file /etc/grid-security/grid-mapfile để xác định người dùng địa phương có quyền thực thi công việc do một yêu cầu của một người dùng Grid gửi tới.

2.5 Quản lý dữ liệu trong GT4

2.5.1 Tổng quan về quản lý dữ liệu trong GT4

Các công cụ quản lý dữ liệu trong GT4 liên quan đến việc lưu trữ, vận chuyển và quản lý các dữ liệu phân tán trong môi trường không đồng nhất. GT4 cung cấp nhiều thành phần bao gồm [6]:

- *GridFTP* chịu trách nhiệm vận chuyển dữ liệu tin cậy và hiệu quả
- *RFT* quản lý nhiều quá trình vận chuyển
- *RLS* duy trì thông tin về các bản sao dữ liệu
- *OGSA-DAI* cho phép truy xuất và tích hợp dữ liệu có/bán cấu trúc

Ngoài ra còn phải kể đến các công cụ kết hợp khác như:

- *NeST* giải quyết việc đặt chỗ để lưu dữ liệu
- *UberFTP* cung cấp client dòng lệnh cho GridFTP
- *SRB* cung cấp giao diện chung cho dữ liệu phân tán
- *DataCutter, STORM* hỗ trợ xử lý pipeline cho dữ liệu phân tán
- *SRM – Storage Resource Manager* cung cấp truy xuất GridFTP trên nền bảo mật GSI tới các kho lưu trữ
- *GridNFS* cài đặt *NFSv4* trên nền GSI phát triển bởi đại học Michigan

Các thành phần chính của GT4 chia thành 2 nhóm chính: di chuyển dữ liệu và sao lưu dữ liệu.

2.5.2 Di chuyển dữ liệu

Có 2 thành phần liên quan đến vận chuyển dữ liệu, đó là GridFTP và RFT.

- ***GridFTP***

Giao thức này trong GT4 cung cấp khả năng chuyển dữ liệu an toàn, nhanh, mạnh, hiệu quả và cho phép truyền file theo mô hình third-party. Trong mô hình truyền file truyền thống, việc truyền file chỉ thực hiện được giữa client và server. Còn trong mô hình truyền file third-party, Client có thể thực hiện truyền file giữa các server với nhau.

Bộ Globus Toolkit cung cấp gần như hoàn chỉnh cài đặt của giao thức này bao gồm: *GridFTP globus-gridftp-server*, lệnh *globus-url-copy* và thư viện lập trình. Để chia sẻ dữ liệu ta cài đặt một GridFTP server trên máy chủ mà có thể truy xuất được dữ liệu, và một giao diện lưu trữ dữ liệu Data Storage Interface (DSI) phù hợp với hệ thống đang lưu trữ dữ liệu. Để truy xuất đến dữ liệu đã được chia sẻ, ta cần một GridFTP client. GT cung cấp một lệnh client là globus-url-copy. Lệnh này cho phép truy xuất dữ liệu qua nhiều giao thức (http, https, ftp, gsiftp, và file).

- ***Dịch vụ chuyển tập tin tin cậy - RFT***

Khi yêu cầu chuyển tập tin được chấp nhận, chúng ta có thể theo dõi trạng thái truyền qua các phương thức mà dịch vụ RFT cung cấp. GT4 cung cấp một cài đặt của dịch vụ RFT trong GT container và một client khá đơn giản. Ngoài ra lập trình viên có thể dùng một số lớp Java để phát triển theo ý muốn.

2.5.3 Tạo bản sao dữ liệu

- ***Dịch vụ định vị bản sao - RLS***

RLS là dịch vụ cho phép người dùng hoặc dịch vụ khác đăng ký tập tin trong RLS khi tập tin được tạo ra, sau đó nó chịu trách nhiệm tạo và duy trì các bảo sao của tập tin này trên các hệ thống lưu trữ phân tán. Người dùng có thể truy vấn RLS server để tìm các bản sao này. RLS gồm nhiều server nằm rải rác lưu dữ liệu đăng ký. Bằng cách phân tán dữ liệu, kích thước của hệ thống có thể mở rộng và lưu trữ được nhiều bản sao hơn hệ thống quản lý tập trung.

Trong GT4, nhiệm vụ của RLS là duy trì các ánh xạ giữa tên tập tin logic với một hoặc nhiều tên tập tin vật lý. Người dùng có thể yêu cầu danh sách các tập vật lý của một tên tập tin logic và ngược lại. Ngoài ra, RLS cho phép gắn các thuộc tính

hoặc thông tin mô tả (kích thước hoặc checksum) với các tên tập tin logic hoặc vật lý, và cho phép truy vấn theo những thuộc tính này.

- ***Dịch vụ sao dữ liệu – DRS***

GT4 đã thiết kế và cài đặt dịch vụ DRS dựa trên 2 thành phần quản lý dữ liệu của GT là RFT và RLS. Chức năng của DRS là đảm bảo một tập các files tồn tại trên một nơi lưu trữ. Đầu tiên, dịch vụ DRS sẽ truy vấn RLS server để xác định vị trí của các file mong muốn trong mạng lưới. Sau khi định vị xong, DRS sẽ yêu cầu chuyển tập tin tới nơi mong muốn bằng dịch vụ RFT. Cuối cùng, DRS sẽ đăng ký bản sao mới này với RLS server.

DRS được cài đặt như một dịch vụ Web tuân theo cơ cấu WSRF. Khi nhận một yêu cầu DRS, một WS-Resource được sinh ra để lưu trạng thái cho mỗi file cần sao chép, bao gồm cả thao tác trên file thành công hoặc không.

2.6 Theo dõi và phát hiện

Các cơ chế theo dõi và phát hiện nhằm giải quyết vấn đề liên quan đến thu thập, phân tán, chỉ mục hóa, lưu trữ và xử lý thông tin về cấu hình và trạng thái của dịch vụ và tài nguyên. GT4 cung cấp:

1. Dịch vụ Web cơ bản cho các giao diện WSRF và WS-Notification. Đây là nền tảng để xây dựng các dịch vụ theo dõi và phát hiện, để định nghĩa các thuộc tính cần thiết cho các dịch vụ này, và hỗ trợ chế độ truy xuất kéo/đẩy.
2. Dịch vụ GRAM và RFT định nghĩa các Resource Properties làm cơ sở cho dịch vụ theo dõi và phát hiện.
3. Dịch vụ chỉ mục hóa (MDS-Index), lưu trữ (MDS-Archive), và phân tích dữ liệu đối với các sự kiện (MDS-Trigger).
4. Mỗi GT container chứa dịch vụ MDS-Index để phát hiện các dịch vụ trong container cũng như để liên kết với các chỉ mục khác trên mạng lưới.

2.6.1 Hệ thống theo dõi và phát hiện - MDS4

Hệ thống theo dõi và phát hiện MDS4 là một thành phần của GT4 thực hiện theo dõi và phát hiện các dịch vụ, tài nguyên trong hệ thống phân tán. *Monitoring* là quá trình thu thập thông tin về các tài nguyên hoặc dịch vụ, nhằm khắc phục lỗi và

theo dõi quá trình hoạt động. Người dùng có thể dùng hệ thống theo dõi này để nhận biết tài nguyên đang cạn kiệt để tìm giải pháp khắc phục kịp thời. *Discovery* là quá trình tìm kiếm tài nguyên thích hợp để thực hiện một công việc, ví dụ như: tìm một máy chủ tính toán để thực thi một job. Quá trình này bao gồm việc tìm tài nguyên thích hợp và lựa chọn cái nào phù hợp nhất.

Những ứng dụng theo dõi và phát hiện đều đòi hỏi khả năng thu thập thông tin từ nhiều nguồn phân tán. Để thực hiện điều này, MDS4 cung cấp:

- Các dịch vụ kết hợp dùng thu thập thông tin trạng thái từ các nguồn đã được đăng ký
- Giao diện dựa trên trình duyệt, các lệnh, và giao diện dịch vụ Web cho phép người dùng truy vấn thông tin thu thập được
- MDS4 cung cấp 3 dịch vụ kết hợp khác nhau
- MDS-Index hỗ trợ truy vấn Xpath
- MDS-Trigger thực hiện hành động do người dùng mô tả (gửi email hoặc ghi vào log file) khi thông tin nhận được phù hợp với yêu cầu người dùng đặt ra
- MDS-Archiver lưu trữ thông tin trong cơ sở dữ liệu

2.6.2 Bộ gộp (aggregator) và nguồn thông tin

Để hiểu hơn về MDS4 ta sẽ tìm hiểu cơ cấu bộ gộp-nguồn thông tin (aggregator-information source). Ý tưởng chính của cơ cấu này là:

- Các nguồn thông tin cần thiết sẽ được đăng ký với một dịch vụ gộp
- Việc đăng ký có hiệu lực trong một khoảng thời gian. Vì thế bộ gộp gắn với nó sẽ tự huỷ nếu đăng ký không được làm mới
- Bộ gộp liên tục cập nhật trạng thái hiện thời của nguồn thông tin đã đăng ký qua cơ chế truy xuất theo đặc tả nguồn thông tin (information-source-specific access mechanism)
- Sau đó thông tin này sẽ được bộ gộp công bố ra ngoài thông qua giao diện dịch vụ Web theo đặc tả của bộ gộp (aggregator-specific Web services interface)

Các bộ gộp MDS4 hoạt động khác với các phương pháp đăng ký tĩnh truyền thống như UDDI ở chỗ chúng cho phép đăng ký *softstate* các nguồn thông tin và luôn làm tươi các giá trị nguồn thông tin theo chu kỳ.

2.6.3 Nguồn thông tin và việc đăng ký

Việc đăng ký được thực hiện thông qua thao tác *Add* của dịch vụ Web WSServiceGroup. Có 2 chế độ đăng ký tùy theo cơ chế dùng để truy xuất:

1. Chế độ đăng ký tổng quát: cho phép nhận thông tin từ một nguồn bất kỳ. Một nguồn thông tin được đăng ký thông qua một chương trình của người dùng chạy định kỳ để cung cấp thông tin cho bộ gộp. Chương trình này có thể sinh ra thông tin một cách cục bộ hoặc dùng giao thức theo đặc tả nguồn để truy xuất thông tin từ xa, sau đó chuyển đổi dữ liệu non-XML thành XML.

2. Đăng ký qua dịch vụ Web theo cơ cấu WSRF: thông tin trạng thái được biểu diễn như là các WS-Resource properties.

2.7 Kết chương

Như vậy, Globus là bộ công cụ rất hữu dụng để triển khai một lưới. Nó là hạ tầng phần mềm cung cấp các công cụ để xây dựng tính toán mạng lưới và các ứng dụng khác dựa trên công nghệ lưới một cách dễ dàng. Các công cụ này được gọi chung là Globus Toolkit, gồm nhiều module công cụ, mỗi công cụ định nghĩa một giao diện phục vụ ở cấp cao và cung cấp những toán tử ở cấp thấp có thể thực hiện được trên nhiều môi trường khác nhau. Hầu hết các lưới đang được triển khai hiện nay đều được thực hiện dựa trên nền Globus Toolkit.

CHƯƠNG 3. CÁC KỸ THUẬT LƯỚI HIỆN ĐƯỢC TRIỂN KHAI Ở VIỆT NAM

Trong chương này, ta sẽ đi tìm hiểu chi tiết về một số kỹ thuật xây dựng các lưới đã và đang được một số nhóm nghiên cứu về tính toán lưới của nước ta triển khai trong thời gian gần đây.

3.1 Desktop Grids

3.1.1 Tính toán phân tán trong các xí nghiệp

Nhu cầu tính toán phân tán trong các tập đoàn làm xuất hiện các lưới tính toán được tạo ra nhờ kết nối nhiều máy tính chạy hệ điều hành Windows hoặc Unix với nhau. Người sử dụng mong đợi việc quản lý tài nguyên và bảo mật trong lưới cũng như khả năng sử dụng lưới được thiết lập bằng cách kế thừa hạ tầng tính toán phân tán sẵn có của họ.

Tuy nhiên, với dạng lưới tính toán này, mối quan tâm được tập trung chủ yếu vào bốn khía cạnh đó là:

- Quản lý tài nguyên-Resource Management
- Bảo mật dữ liệu và giao tiếp-Communication and Data Security
- Tính không đồng nhất của các máy trong lưới-Machine Heterogeneity
- Tính sẵn dùng của tài nguyên-Resource Availability

Với Desktop Grid, người xây dựng ra nó cố gắng khai thác những khác biệt cơ bản giữa một tập các PC được kết nối trong một mạng công ty với một tập các máy PC đặt tại nhà được kết nối bằng đường Internet bao gồm:

- *Việc kết nối mạng- Network Connectivity*: nhiều tập đoàn có các máy PC được kết nối vào các mạng có tốc độ cao, đạt 100 Mbps hay 1Gbps. Tuy nhiên, với các máy tính chỉ thỉnh thoảng mới kết nối vào mạng của tập đoàn thì khoảng thời gian và chất lượng kết nối thường rất khó định trước
- *Sự tham gia được yêu cầu-Required Participation*: việc tham gia vào lỗ lực tính toán phân tán có thể là một phần yêu cầu của một công ty. Nhờ đó nó cung cấp tính đảm bảo cao hơn về thành phần của lưới

- *Bảo mật và Quản trị các PC-PC Administrator and Security*: đây thực sự là vấn đề quan trọng trong lưới vì những thông tin nhạy cảm có thể được phân bổ trên các nút tính toán của lưới
- *Truy nhập đến các tài nguyên dùng chung-Access to Shared Resource*: hầu như tất cả các tổ chức đều có kho dữ liệu chung trên mạng nội bộ của họ. Đây có thể chỉ đơn giản là một ánh xạ ổ đĩa dùng chung được thiết lập qua một kết nối mạng hoặc phức tạp như một kho dữ liệu với dung lượng lên đến muti-terabyte và multi-tier

3.1.2 Định nghĩa Desktop Grid

Destop Grid là một lưới có các đặc điểm sau:

- Gồm một tập các máy kết nối không liên tục và máy dùng chung được đặt tên trên một một mạng chia
- Gồm một tập chính sách được điều khiển bởi người dùng để mô tả cách mỗi máy tham gia vào lưới
- Một topo mạng ảo được kiểm soát bởi một máy server chuyên dụng. Các máy trên lưới thông báo với nhau thông qua một server trung tâm
- Một cơ chế quản lý chủ động đối với việc phân phối, thực thi và thu hồi công việc đến và đi khỏi lưới nằm dưới sự điều khiển của một server trung tâm

Mục đích phía sau việc hình thành một Desktop grid là kết hợp nhiều tài nguyên lại thành một tài nguyên ảo duy nhất, dễ sử dụng và có thể quản lý được.

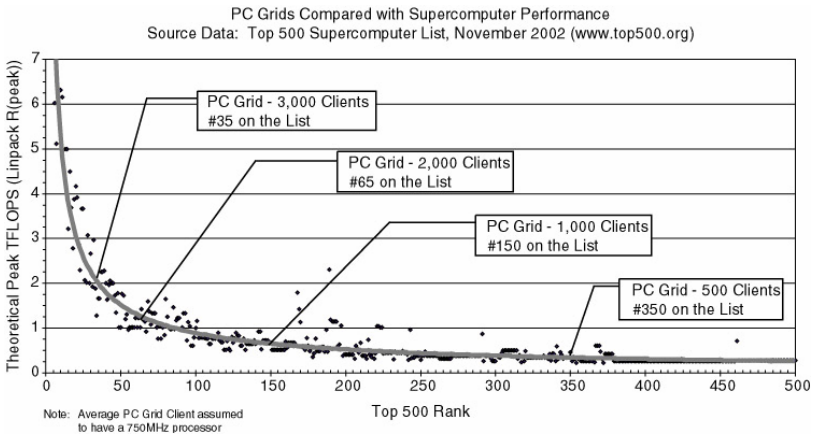
Một số định nghĩa chung cho các thành phần trong Desktop Grid như sau:

- **Grid**: là thuật ngữ được dùng thay cho cụm từ Desktop Grid
- **Grid Server**: là một máy trung tâm dùng để điều khiển và quản trị lưới
- **Grid Client**: là một nút riêng lẻ, thường là một máy PC để bàn hoặc một máy xách tay có sẵn
- **Grid Client Executive**: là một thành phần phần mềm của hạ tầng lưới, cư trú trên một PC, nó có thể làm cho một PC phục vụ như một Grid Client, và quản lý tất cả các tương tác giữa Grid Client và Grid Server

- **Work Unit**: là đơn vị công việc được Grid Server giao cho một Grid Client. Bao gồm một phiên bản có thể chạy trên lưới của ứng dụng, các chỉ dẫn để thiết lập một môi trường cho ứng dụng trên Grid Client, dữ liệu vào, và các chỉ dẫn để thực thi ứng dụng cũng như tạo dữ liệu ra

3.1.3 Giá trị của lưới Desktop Grid

Desktop Grid tạo ra cơ hội gom tài nguyên tính toán rảnh của các PC; đây là một giải pháp hiệu quả để tăng sức mạch tính toán cho tổ chức. Một Desktop Grid đôi khi được xem như một siêu máy tính ảo "virtual supercomputer" Thực tế quan niệm này cũng không sai là mấy. Trong hình dưới, có thể thấy biểu đồ của top 500 siêu máy tính hàng đầu thế giới được tổng kết hồi tháng 11 năm 2002 [3].



Hình 3-1 Xếp hạng của các Desktop Grid trên 500 siêu máy tính hàng đầu

3.1.4 Các phần tử kỹ thuật chính

Các phần tử kỹ thuật chính cần được đánh giá đối với một Desktop Grid là:

- ***Kỹ thuật bảo mật***

Không cho phép ứng dụng phân tán truy xuất đến tài nguyên mạng hoặc tài nguyên cục bộ bằng cách: Mã hoá ứng dụng và dữ liệu; Đảm bảo phần môi trường Grid Client còn lại không bị thay đổi sau khi chạy một ứng dụng phân tán; Ngăn người dùng cục bộ gây phiền phức đến sự thực thi của ứng dụng phân tán đang chạy

trên máy của họ; Ngăn người dùng cục bộ lục lọi và sửa đổi hoặc xoá dữ liệu đi kèm các ứng dụng phân tán.

- ***Kỹ thuật kín đáo***

Grid Client Executive phải tự động giải phóng các tài nguyên khi không còn dùng; Đảm bảo việc thao tác là trong suốt với người dùng địa phương; Ngăn các ứng dụng phân tán hiển thị hộp thoại thông báo hoặc hiển thị các yêu cầu tương tác; Ngăn chặn hiện tượng giảm hiệu năng (hoặc gây lỗi) trên máy client do việc thực hiện ứng dụng phân tán; Không được yêu cầu tương tác với người dùng trên Grid Client.

- ***Kỹ thuật tích hợp ứng dụng***

Kỹ thuật tích hợp ứng dụng yêu cầu: Khả năng mô phỏng một môi trường chuẩn với một Grid Client; Tích hợp mức Binary (không biên dịch lại, không kết nối lại, không truy cập đến mã nguồn); Tích hợp một cách dễ dàng; Bảo mật tích hợp, mã hoá các dữ liệu nhạy cảm và phải hỗ chợ bất cứ ứng dụng Win 32 nào.

- ***Kỹ thuật Robustness***

Kỹ thuật Robusness phải đảm bảo được việc: Tự động tái chỉ định các đơn vị công việc khi các Grid Client bị xoá khỏi lưới; Tự động tái chỉ định các đơn vị công việc do lỗi tài nguyên hoặc lỗi mạng; Ngăn các ứng dụng khác chi phối tài nguyên Grid Client; Cung cấp khả năng hỗ trợ trong suốt cho tất cả các phiên bản Windows khác nhau trong Grid Client.

- ***Kỹ thuật co dãn***

Một lưới Grid Desktop phải có thể: tự động thêm, cấu hình, đăng ký các Grid Client mới; tương thích với nhiều tài nguyên khác nhau; có thể cấu hình trên nhiều vị trí địa lý.

- ***Kỹ thuật quản lý tập trung***

Chính là việc giám sát tài nguyên lưới tự động; Quản lý và xếp hàng các đơn vị công việc tập trung; Quản trị với chính sách tập trung đối với việc truy xuất và sử dụng lưới; Tương thích với các hệ thống quản lý IT đang tồn tại; Việc khởi tạo và

cập nhật sản phẩn có thể được hoàn thành bằng cách sử dụng các môi trường quản lý phần mềm xí nghiệp như SMS, WinInstall; và Triển khai, quản lý client từ xa.

3.1.5 Các khía cạnh thực tế cần xem xét

Là các khía cạnh thực tế chính nảy sinh khi tích hợp máy để bàn chuyên dùng trong xí nghiệp thành một lưới.

- ***Các ứng dụng***

Các ứng dụng chạy trên lưới để bàn thường được chia thành 3 nhóm là:

- *Data Parallel*: xử lý ctập dữ liệu vào lớn, liên tục và không lệ thuộc nhau
- *Parameter Sweep*: sử dụng phương thức tương tác để tạo ra một chuỗi các giá trị đầu vào nhiều
- *Probabilisti*: xử lý một số thử nghiệm sử dụng nhập liệu ngẫu nhiên để tạo ra các giá trị đầu vào dùng để định giá một tập các chức năng output

Phân tích khả năng phân tán của ứng dụng

Hiểu tham số nào của ứng dụng là hard-code và tham số nào là có thể thay đổi được dựa trên các giá trị đầu vào. Tiếp đến ta phải hiểu cách phân tích đầu vào của một công việc lớn thành một tập các đầu vào nhỏ hơn có thể xử lý theo kiểu phân tán và hiểu cách tập hợp kết quả đầu ra từ thành kết quả cuối cùng.

Xác định tính phù hợp của ứng dụng

Tính thích hợp của ứng dụng thường được đánh giá thông qua tỉ lệ Compute Intensity (CI) được tính như sau:

CI = 4*WorkUnitDuration (giay)/(**InputSize**(KB) + **OutputSize**(KB))

Trong đó WorkUnitDuration là khoảng thời gian thực hiện một đơn vị công việc. Thông thường, nếu một ứng dụng có thể lưới hoá có CI>1.0 thì thường được cho là có thể thực hiện xử lý phân tán bằng giải pháp lưới để bàn. Nhưng đôi khi vẫn có một số ngoại lệ.

- ***Môi trường tính toán***

Desktop Grid được coi như một tuỳ chọn có chi phí thấp và có thể được dùng để thay cho các tài nguyên tính toán mới. Trong bất cứ trường hợp nào, việc thêm

một Desktop Grid vào một tổ chức là một thay đổi đáng kể đối với toàn bộ môi trường tính toán.

- ***Văn hoá***

Đầu tiên là văn hoá của nhân viên, anh ta phải biết được vai trò của PC mà anh ta đang sử dụng có tầm quan trong ra sao đối với thành công của cả tổ chức khi tham gia vào lưới. Họ phải có kiến thức nhất định về Grid Client Executive để không thực hiện những hành đồng sai lầm dẫn đến gây ra lỗi trên PC của họ hoặc để lây nhiễm virus cũng như các chương trình gián điệp trên máy…

3.1.6 Grid Server

Grid Server cung cấp khả năng quản lý/quản trị toàn bộ đơn vị công việc, gán các đơn vị công việc cho Grid Client, quản lý/quản trị Grid Client. Ngoài ra, nó còn có thể cung cấp một số chức năng khác như:

- *Các thao tác mức nhóm các Client* (Client Group-level Operations): khi khích thước của lưới tăng lên sẽ chia nhóm nhỏ để quản lý
- *Data Caching*: thời gian được yêu cầu để truyền dữ liệu đi hoặc đến các Client đóng một vai trò khá quan trọng trong việc tính toán tỉ số CI
- *Quản trị và lập lịch mức công việc* (Job-level Scheduling and Administration): hỗ trợ nhiều mức ưu tiên công việc khác nhau theo khả năng để chọn lựa Client hoặc nhóm Client
- *Điều chỉnh và phân tích hiệu năng* (Performance Tuning and Analysis): cung cấp các báo cáo và dữ liệu cần thiết để xác định được những đặc trưng hiệu năng quan trọng của mỗi Grid Client cũng như của cả lưới
- *Bảo mật* (Security): mỗi chức năng được định nghĩa riêng biệt trên Grid Server, môi trường người dùng nên bao gồm cả bảo mật mức người dùng
- *Các giao diện hệ thống* (System Interfaces): hỗ trợ một tập các giao diện khác nhau cho nhiều chức năng người dùng và chức năng quản trị khác nhau

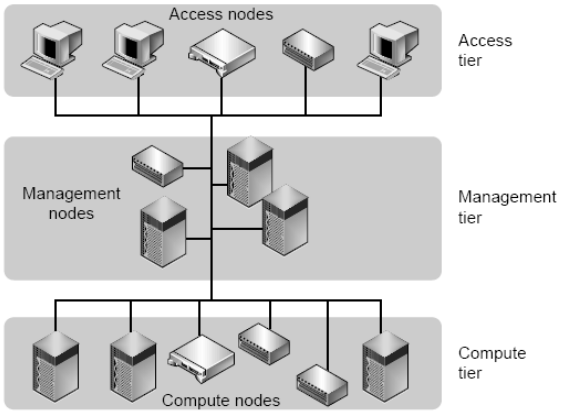
3.2 Cluster Grids

Một lưới cluster là dạng đơn giản nhất của Grid, nó cung cấp dịch vụ tính toán cho một nhóm hoặc một phòng. Kiến trúc lưới dạng cluster giúp cực đại hoá khả năng sử dụng các tài nguyên và tăng thông lượng cho công việc của người dùng thông qua việc ảo hóa tài nguyên tính toán.

Phần này ta đi tìm hiểu về kỹ thuật Cluster Grid với trường hợp cụ thể là các Cluster Grid của Sun [13].

3.2.1 Kiến trúc lưới Cluster

Kiến trúc của lưới cluster được chia thành 3 lớp logic [13] không phân cấp bao gồm: lớp truy xuất, lớp quản lý và lớp tính toán.



Hình 3-2 Ba lớp trong kiến trúc lưới cluster

- ***Lớp truy xuất***

Lớp này cung cấp các dịch vụ truy xuất, xác minh quyền cho người dùng lưới. Các lệnh truy xuất thông thường như *telnet*, *rlogin*, *ftp*, *ssh* để truy nhập vào hệ thống và các dịnh vụ dựa trên nền Web.

- ***Lớp quản lý***

Đây là lớp trung gian, chịu trách nhiệm cung cấp các dịch vụ chính trong lưới cluster như: DRM, phần mềm chuẩn đoán phần cứng tự động, kiểm soát hiệu năng của hệ thống…Trong đó DRM là dịch vụ bắt buộc phải có của một lưới

cluster, ngoài ra lớp này còn có thêm một số nhiệm vụ khác như: Dịch vụ file; Dịch vụ khóa bản quyền; Quản lý sao lưu; Dịch vụ cài đặt.

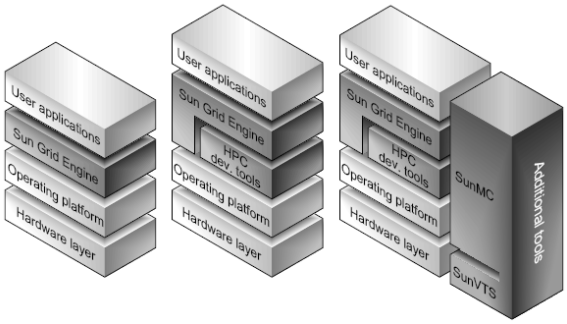
- ***Lớp tính toán***

Lớp tính toán cung cấp sức mạnh tính toán cho lưới cluster. Các công việc đệ trình thông qua các lớp bên trên trong kiến trúc này được xếp lịch để chạy trên một hoặc nhiều nút trong lớp tính toán. Các nút trong lớp tính toán chạy các tiến trình phía khách hoặc tác tử của phần mềm DRM, và các tác tử để giám sát sức khỏe của hệ thống. Lớp tính toán giao tiếp với lớp quản lý thông qua việc nhận các công việc, báo cáo trạng thái hoàn thành công việc và các chi tiết tính toán.

Lớp tính toán có thể là sự lai ghép của nhiều loại máy phục vụ, nhiều nền hệ điều hành khác nhau. Ngoài ra ở lớp này còn có sự khác nhau về nhóm các chức năng trong mỗi nút tính toán và cả cách mỗi nút này được gắn vào lưới.

3.2.2 Bó phần mềm lưới cluster của Sun

Bó phần mềm lưới cluster của Sun bao gồm một số các thành phần chính là Sun Grid Engine (SGE) software và Sun HPC ClusterTools™ software [13].



Hình 3-3 Ngăn xếp phần mềm của lưới Sun Cluster Grid

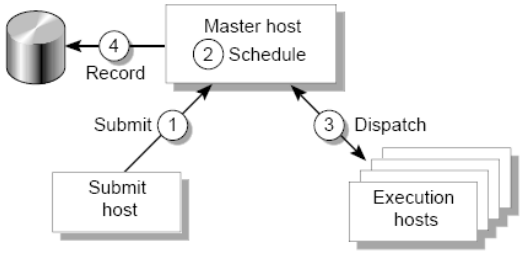
- ***Sun Grid Engine - SGE***

Phần mềm quản lý tài nguyên phân tán SGE là thành phần cốt lõi, nó cung cấp tất cả các chức năng quản lý tài nguyên truyền thống, chẳng hạn như xếp hàng theo lô, cân bằng tải, thống kê tài khoản công việc, quản lý các tài nguyên hoặc người dùng đặc biệt, treo hoặc phục hồi các công việc và quản lý tài nguyên trên toàn cluster.

Host	Mô tả
Master	Xử lý các yêu cầu đến từ phía user (tiến trình Qmaster), thực hiện xếp lịch công việc và gửi chúng tới các host thực hiện (tiến trình Schedd). Có một Master Host chủ duy nhất trong mỗi phần mềm SGE.
Shadow master	Liên tục giám sát master host chính (tiến trình Shadowd) một cách trong suốt và tự động, nó sẽ chiến quyền kiểm soát của master host này trong trường hợp nó bị lỗi
Execution	Là host thích hợp nhất để thực thi các công việc. Tiến trình Execd nhận công việc từ Qmaster và sinh ra một tiến trình dẫn để thực hiện công việc này trên local. Thông báo thông tin ngược lại cho Qmaster.
Submit	Các host này được cấu hình để đệ trình, giám sát và quản trị các công việc. Không có tiến trình ngầm nào được yêu cầu.
Admin	Là các host tạo thay đổi cấu hình của cluster, chẳng hạn như là thay đổi các tham số DRM như thêm các nút mới hoặc user mới. Không có tiến trình ngầm nào được yêu cầu cho các host dạng này.

Luồng công việc

Tất cả các công việc được đệ trình tới SGE master và được lưu trong spooling area cho đến khi bộ xếp lịch quyết định xem công việc nào được phép thực hiện. Phần mềm SGE nối các tài nguyên có thể tới các yêu cầu công việc. Ngay khi tìm được một tài nguyên thích hợp cho công việc mới, SGE chuyển công việc này thành công việc ưu tiên cao nhất. Sau đó bộ xếp lịch SGE đặt chúng vào các máy có thể thực hiện.



Hình 3-4 Luồng công việc trong Sun Grid Engine

1. Công việc đệ trình từ một submit host, yêu cầu được gửi đến master host.

2. Master host xác định host sẽ thực thi công việc, đánh giá mức độ công việc, kiểm tra giấy phép bản quyền và định giá các yêu cầu công việc khác.

3. Master host gửi công việc tới execution host được chọn. Execution host lưu lại công việc trong một job information database và bắt đầu một tiến trình dẫn, và đợi nó hoàn thành.

4. Khi công việc được hoàn thành, tiến trình dẫn trả về thông tin công việc và execution host sau đó sẽ thông báo việc hoàn thành công việc tới master host, đồng thời xóa công việc khỏi job information database. Master host cập nhật job accountisang database thành hoàn thành công việc.

- ***Sun HPC ClusterTools Software***

Sun HPC ClusterTools software cung cấp môi trường phần mềm tích hợp cho các ứng dụng MPI. Phần mềm Sun HPC ClusterTools là các ứng dụng song song lai rất linh hoạt và an toàn, nó được trộn giữa kỹ thuật tuyến đoạn và song song để tạo ra các ứng dụng sử dụng MPI để giao tiếp giữa các tiến trình.

Ngoài các phần mềm chính trên, còn có thêm các phần mềm khác như:

- ***Solaris JumpStart và Solaris Flash Software***

Là công cụ để tăng tốc và khởi tạo tự động trong môi trường hệ điều hành Sun. Solaris JumpStart là một chương trình cài đặt tự động môi trường hệ điều hành Solaris tại bất cứ vị trí nào trên mạng. Solaris Flash bắt giữ một ảnh hệ thống, bao gồm môi trường hệ điều hành Solaris, hầu hết các ứng dụng, và thông tin cấu hình hệ thống thông qua việc sử dụng flarcreate command (flash tool) giúp quản trị viên có thể lặp lại các cấu hình máy chủ trên nhiều máy chủ khác.

- ***Phần mềm Quản lý tập trung - SunMC***

Là một phần mềm quản lý hệ thống giúp nhà quản trị hệ thống thực hiện việc quản lý hệ thống từ xa, điều khiển và cách lý phân cứng khỏi các lỗi phần mềm trên các hệ thống Sun. Nó có 3 thành phần: *Console* cung cấp giao diện người dùng dưới dạng GUI hoặc giao diện trình duyệt web. *Máy chủ SunMC* giám sát và điều khiển các thực thể được quản lý bằng cách gửi yêu cầu tới các tác tử trú ngụ trên các nút được quản lý. *Các tác tử* là các thành phần phần mềm dựa trên giao thức SNMP, nó

thu thập dữ liệu quản lý nhân danh máy chủ, sau đó gửi về cho máy chủ và chạy các tiến trình.

➢ ***Phần mềm phê chuẩn kiểm tra tính phù hợp***

Phần mềm SunVTS là một công cụ chuẩn đoán, kiểm tra và phê chuẩn phần cứng Sun thông qua việc xác minh tính liên thông và chức năng của hầu hết các thiết bị phần cứng trong hệ thống Sun.

3.2.3 Yêu cầu thiết kế

Trong phần này, các thông tin được trình bày đi sâu vào các giới thiệu và các yêu cầu cho một thiết kết lưới cluster của Sun.

➢ ***Lớp truy xuất***

Ở mức tối thiểu, lớp truy xuất phải có thể hỗ trợ việc đăng nhập, hỗ trợ các phiên telnet và đệ trình công việc. Các công việc không tương tác được thực hiện thông qua hệ tống hàng đợi bằng cách đệ trình một một batch shell script đơn giản. Nếu chỉ một hệ thống được yêu cầu để hỗ trợ hàng trăm đăng nhập từ xa, hệ thống này cần được đảm bảo chắc chắn một số tài nguyên dành riêng. Việc truy xuất của người quản trị vào lưới được cung cấp thông qua phần mềm Sun Grid Engine và SunMC qua một giao diện dòng lệnh hoặc một giao diện đồ họa. Nếu các truy nhập dựa trên web phải được thực hiện, thì máy chủ phải được ước lượng phù hợp để thực hiện được yêu cầu.

➢ ***Lớp quản lý***

Đối với các lưới cluster nhỏ, cung cấp dịch vụ tối thiểu (chỉ có DRM), thì một bộ xử lý đơn là đủ. Nếu một máy đa bộ xử lý được triển khai trong lưới cluster với vai trò là SGE master thì nó có thể dùng để cung cấp các dịch vụ khác như Sun MC server, NFS and backup server đồng thời cũng có thể hỗ trợ các tác vụ tính toán.

Các phần tử khác nhau trong một lưới có thể yêu cầu một hệ thống file chia sẻ là: phần mềm Sun Grid Engine, thư mục cá nhân của người dùng, cặp Sun HPC

ClusterTools™ 4.0 , các máy chủ ứng dụng, máy chủ kiểm tra khoá bản quyền và các máy chủ cài đặt.

➢ ***Lớp tính toán***

Việc quyết định chọn phần cứng nào cho lớp tính toán phụ thuộc chủ yếu vào hiệu năng cực đại có thể thu được trên cùng lượng tiền bỏ ra. Một số điểm nên chú ý khi chọn là:

- *Thông lượng*: số lượng các nút đơn trong mạng. Mong muốn cực đại hoá số các bộ xử lý trên mỗi volume hơn là hiệu năng trên từng bộ xử lý đơn
- *Tính song song cao*: phụ thuộc vào các thuộc tính của ứng dụng, hoặc một lượng nhỏ nút SMP lớn hoặc một lượng lớn các nút yếu được hỗ trợ bởi một môi trường cluster thời gian chạy (CRRE) sẽ là thích hợp
- *Các yêu cầu về bộ nhớ và bộ nhớ đệm*: một số ứng dụng thu được kết quả rất khả quan từ các bộ xử lý có bộ nhớ đệm lớn. Trong trường hợp như thế mục đích là nhằm cực đại hoá tính cân xứng của dữ liệu đang active, dữ liệu này dược lưu trong các bộ nhó đệm nhanh, giúp cho việc truy nhập dữ liệu tốn ít thời gian hơn là truy nhập vào bộ nhớ lưu trữ thông thường

3.2.4 Phần cứng mạng

Có 3 kiểu liên kết nối có thể được xem xét: các kết nối nối tiếp, kết nối qua Ethernet và các kết nối có độ trễ đặc biệt thấp

➢ ***Nối tiếp***

Một mạng nối tiếp cho phép nhà quản trị hệ thống thực hiện việc truy nhập từ màn hình console tới tất cả các máy trong cluster. Đối với các môi trường lớn, cho phép hoàn thành hầu hết các điều khiển trên toàn bộ hệ thống thông qua việc truy nhập từ xa. Việc sử dụng một máy trạm tập trung giúp nhà quản trị truy nhập tới các cổng console thông qua mạng Ethernet.

➢ ***Ethernet***

Tải của mạng trong một lưới cluster là do các giao tiếp chuyển thông điệp MPI hoặc PVM ở chế độ chạy. Lưu lượng NFS từ rất nhiều nguồn như các truy

nhập các dịch vụ lưới, các file spool, các thư mục người dùng đang được truy nhập tại thời điểm chạy, các giao tiếp Sun Grid Engine và việc truyền dữ liệu do các thao tác sao lưu và khởi tạo. Tải được tạo ra bởi các lưu lượng này được xử lý rất tốt trên mạng Ethernet chuẩn hoặc các mạng Ethernet gigabit.

- ***Các liên kết nối đặc biệt***

Mạng Myrinet được kết hợp từ nhiều giao diện, kết nối và nhiều chuyển mạch. Đối với các máy Sun, giao diện PCI được dùng với các driver được thay đổi để dùng trong kiến trúc Sun, có thể lấy từ Mycricom. Mycricom hỗ trợ một module giao thức có thể tải được (PM) cho phần mềm Sun HPC ClusterTools 4.0 software. Các PMs được sử dụng bởi phần mềm Sun HPC ClusterTools để mang lưu lượng giữa các tiến trình để khai thác độ trễ thấp và tỉ lệ dữ liệu cao của Myrinet. Liên kết nối Myrinet có thể làm giảm độ trễ thông qua việc xắp thứ tự theo độ lớn và có thể đưa ra lượng băng thông trung bình cao hơn cho các cluster tính toán

3.2.5 Quản lý một Cluster Grid

Trong nhiều môi trường lưới cluster, một lượng lớn các hệ thống được cấu hình tương tự nhau đang tồn tại, nó giúp tối thiểu hoá thời gian quản trị bằng cách lấy những điểm thuận tiện của các công cụ cài đặt và chuẩn đoán tự động.

- ***SunVTS Software***

Phần mềm SunVTS có thể được dùng để thực hiện kiểm tra phần cứng trên các nút mới trước khi các nút này được đẩy vào lưới để sử dụng. Nó cũng có thể được dùng cho các thao tác kiểm tra phần cứng trong khi chạy, hoặc dùng để điều tra nghiên cứu một nút đang thao tác.

- ***Cluster Console Manager***

Trong môi trường tính toán dạng compute farm environment, các tác vụ yêu cầu việc nhập lệnh như nhau tới nhiều clients, chẳng hạn như đường dẫn cài đặt và các bản update phần mềm, có thể được thực hiện bằng cách sử dụng bộ công cụ Cluster Console được cung cấp kèm với phần mềm Sun HPC ClusterTools. Phần mềm The Cluster Console Manager (CCM) giúp bạn có thể đẩy các câu lệnh tới các

nút trong lưới cùng một lúc thông qua một giao diện đồ hoạ. CCM đề cập đến 3 mô hình thực hiện: *console*, *ctelnet* và *crlogin*

➢ ***Solaris Jumpstart và Flash Software***

Phần mềm Solaris Jumpstart thường được dùng để thực hiện các cài đặt ít nhất là cho lớp tính toán của lưới cluster. Nếu các host mới được thêm vào trong lớp này, hoặc có các nút cần được cài đặt lại thì một môi trường Jumpstart được cấu hình tốt sẽ làm giảm đáng kể thời gian quản lý cho các tác vụ kiểu này

➢ ***Sun Grid Engine Software***

Khả năng đăng nhập và định tài khoản của Sun Grid Engine cho phép nhà quản trị giữ được vết các công việc tính toán đang chạy trên lưới cluster. Phần mềm này lưu một bản ghi cho mỗi công việc được chạy thông qua SGE, bao gồm cả những thông tin chi tiết như thời gian bắt đầu chạy, khoảng thời gian thực hiện, thông tin về CPU, bộ nhớ và I/O, các thông kê người dùng và trong trường hợp SGEEE, nó còn ghi lại cả thông tin về phòng và dự án.

3.3 Kết nối Cluster vào Grid

Để thực hiện các bài toán song song người ta sử dụng một hệ thống gọi là hệ thống tính toán song song phân cụm. Đó là một hệ thống máy tính cục bộ bao gồm một tập các máy tính độc lập và một mạng liên kết giữa các máy tính. Một hệ thống tính toán song song phân cụm thường mang tính chất cục, được quản lý tập trung như là một hệ thống thống nhất.

Hệ thống tính toán song song là hệ thống có bộ nhớ phân tán, truyền thông giữa các nút trong quá trình tính toán thường là truyền thông điệp. Các thư viện truyền thông điệp dành cho hệ thống bó hiện nay có thể kể đến là MPI, PVM, Active Message và CMMD (Connected Machine Message...), trong đó MPI với nhiều ưu điểm có thể coi là sự lựa chọn tốt cho các clusster.

Trong khi các thư viện và các ngôn ngữ lập trình song song tạo cho người lập trình môi trường lập trình trên hệ thống tính toán song song, thì các phần mềm quản lý tài nguyên và phân tải lại tạo ra một môi trường tính toán hiệu quả và ổn

định. Một số phần mềm phân tải và quản lý tài nguyên thường được sử dụng trong các hệ thống tính toán song song hiện nay là: ***PBS (Portable Batch System), CODINE*** và ***CONDOR.*** Ở đây, ta chọn PBS làm phần mềm phân tải và quản lý tài nguyên trong Cluster.

3.3.1 Sự cần thiết của việc kết nối grid và cluster

Trong Chương 1 ta đã có sự so sánh giữa Cluster và Grid, cũng như đã đưa ra điểm giống và khác giữa chúng. Có nhiều ứng dụng lưới rất cần được giải quyết trong phạm vi một tổ chức như cluster chứ không cần được phân bổ trên một tập tài nguyên rộng phân tán về mặt địa lý như lưới. Ngoài ra cluster còn có các đặc điểm mà một lưới thường không có như:

- Nó thuộc về một tổ chức, các tài nguyên là các máy tính đồng nhất, phân bố trên một giới hạn địa lý rất hẹp, được liên kết với nhau bởi các mạng LAN tốc độ cao (cỡ vài trăm Mbits – Gbits) do vậy có độ tin cậy rất cao
- Trong cluster các vấn đề về an ninh không được coi trọng. Vấn đề chủ yếu là làm sao để có được hiệu năng cao và không cần thiết phải hy sinh một phần hiệu năng vì các thủ tục an ninh. Trong cluster người ta có thể làm mọi thứ để đổi lấy hiệu năng
- Cuối cùng cũng cần phải nói thêm rằng tính toán lưới không phải là chìa khóa vạn năng dùng để giải quyết mọi vấn đề. Nó được dùng để bổ trợ chứ không phải là thay thế hoàn toàn các công nghệ tính toán hiện tại

Vì vậy, việc kết nối lưới với cluster đóng vai trò cực kỳ quan trọng, nó sẽ đem lại một sức mạnh tính toán vô cùng to lớn. Với sự kết hợp này ta sẽ tận dụng được sức mạnh sẵn có của các cluster và sử dụng tính toán lưới như là phương tiện để phối hợp các cluster tại các miền địa lý khác nhau cùng giải quyết một bài toán.

3.3.2 Kết nối Globus-based Grid và PBS-based Cluster

3.3.2.1 GRAM

Trong chương 2 ta đã nói khá rõ về thành phần GRAM trong bộ công cụ Globus Toolkit. Do đó, trong mục này ta chỉ tập trung mô tả xem GRAM đóng vai

trò như thế nào? Nó làm gì? Và làm thế nào? khi ta thực hiện kết nối một Cluster vào lưới.

Ta có thể coi GRAM như là thành phần trung gian, với các ứng dụng và dịch vụ mức cao hơn ở phía trên và các cơ chế truy nhập, điều khiển địa phương ở phía dưới. Cả hai phía chỉ cần làm việc với GRAM, bởi vậy sự ảnh hưởng lẫn nhau, các hàm API, và các giao thức cần sử dụng được giảm đi nhiều.

GRAM không cung cấp khả năng lập lịch cũng như môi giới tài nguyên. GRAM cũng không cung cấp các đặc trưng về kế toán và lập hóa đơn. Những đặc trưng được cung cấp bởi các cơ chế quản lý địa phương khác như hệ thống hàng đợi hoặc bộ lập lịch. GRAM làm việc như một giao tiếp trừu tượng cho các nguồn tài nguyên không đồng nhất trên lưới, cụ thể nó chịu trách nhiệm:

- Phân bố tài nguyên cho các công việc
- Đệ trình công việc, chạy ứng dụng trên các máy từ xa và nhận kết quả trả về
- Quản lý trạng thái công việc và các tiến trình

Như vậy có thể thấy được GRAM đóng vai trò như thành phần trung gian làm nhiệm vụ giao tiếp giữa môi trường tính toán toàn cục với các thành phần cục bộ có khả năng tạo ra các tiến trình chạy. Vì vậy, điểm quan trong của GRAM là, nó phải đáp ứng việc giao tiếp với các bộ quản lí tài nguyên địa phương chẳng hạn như: Condor, Fork, PBS, LSF, NQE, v.v... trong đó phần mềm quản lý tài nguyên và phân tải PBS được sử dụng rộng rãi nhất.

3.3.2.2 PBS

- ***Giới thiệu PBS***

PBS (Portable Batch System) [15] [14] là hệ thống phân tải và quản lý tài nguyên, dùng trong các hệ thống tính toán song song phân cụm. Nhiệm vụ chính của hệ thống PBS đó là cung cấp các khả năng về khởi tạo và lập lịch cho việc thực thi các công việc và sắp xếp các công việc đó trên các máy trạm. Hệ thống cũng cho phép xác định trước các nguồn tài nguyên sẽ được sử dụng khi thực thi công việc và đảm bảo chắc chắn dùng đúng tài nguyên yêu cầu.

PBS đảm nhận giao tiếp giữa Server và các máy trạm, máy chủ thông qua PBS quản lý hoạt động của các máy trạm. PBS bao gồm ba môđun chính: PBS_Server, PBS_Scheduler, PBS_MOM. PBS cung cấp rất nhiều chức năng cho các hệ thống tính toán hiệu năng cao. Sau đây là một số chức năng quan trọng nhất của PBS:

1. Chia sẻ tài nguyên: cung cấp một cơ chế lập lịch cho các công việc một cách trong suốt trên bất kỳ một hệ thống PBS nào, bởi bất kỳ người sử dụng nào có thẩm quyền. Công việc có thể được yêu cầu từ một máy client bất kỳ, cục bộ hay từ xa.

2. Giao diện đồ hoạ: giúp người sử dụng chuyển các yêu cầu tính toán ở chế độ lô hoặc chế độ tương tác, truy vấn các công việc, hàng đợi công việc và tình trạng hệ thống; và theo dõi sự tiến triển của các công việc. Ngoài ra PBS cũng hỗ trợ giao diện sử dụng chế độ dòng lệnh cho những người sử dụng chuyên nghiệp.

3. Cơ chế bảo mật: cho phép quản trị thiết lập hoặc huỷ bỏ việc truy cập đến PBS của một người sử dụng, một nhóm người, một máy hoặc một mạng nào đó.

4. Nhật ký: cho phép ghi lại tất cả các hoạt động trên của hệ thống theo từng người, từng nhóm người hoặc từng máy.

5. Tự động chuyển tệp: là cơ chế sao chép các tệp cần thiết cho việc thực hiện một công việc trên các máy trạm tính toán. Các tệp cần chuyển có thể là các tệp dữ liệu hoặc các tệp thực thi.

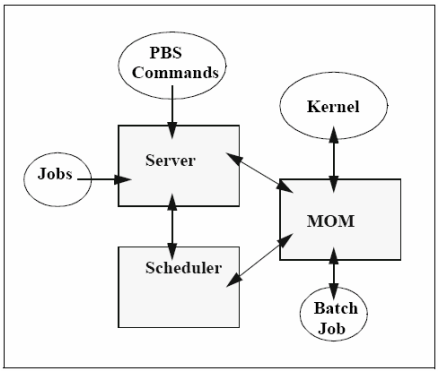
6. Hỗ trợ các công việc song song: cho phép hoạt động cùng với các thư viện lập trình song song như MPICH, MPI-LAM. Các chương trình có thể được lập lịch để chạy trên các hệ đa bộ xử lý hoặc trên các hệ thống đa máy tính.

7. Hỗ trợ tính toán lưới: cung cấp công nghệ siêu tính toán (meta-computing) và tính toán lưới, bao gồm việc hỗ trợ cho GGT (Globus Grid Toolkit). Việc hỗ trợ này chỉ được thực hiện trong phiên bản thương mại PBS Pro.

8. Giao diện lập trình được PBS: cung cấp để người lập trình có thể tự viết các lệnh mới cho PBS, tích hợp PBS vào các ứng dụng của họ hoặc cài đặt các cơ chế lập lịch riêng.

9. Tự động phân tải: là cơ chế cho phép phân tải các công việc trên các tài nguyên của hệ thống Cluster.

- ***Các thành phần cơ bản của PBS***

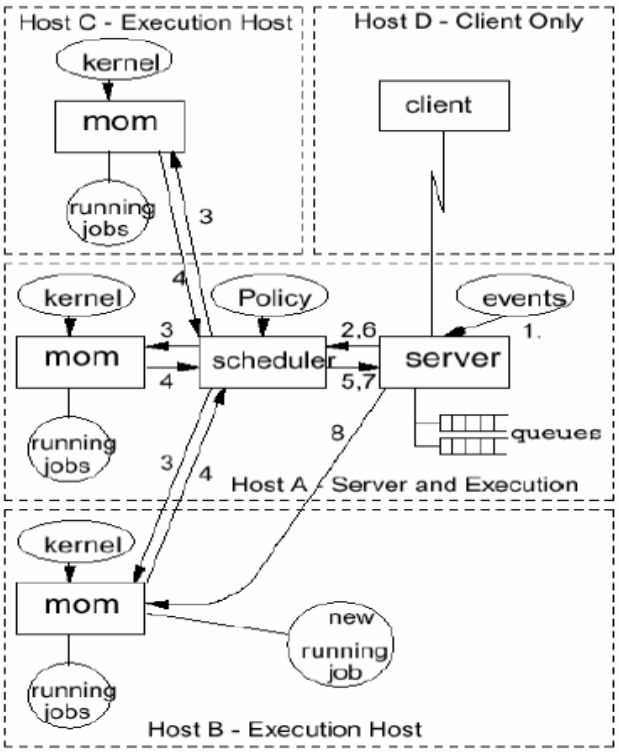


Hình 3-5 Các thành phần trong PBS

1. Mô-đun quản lý công việc (Job Server): là thành phần trung tâm của PBS. Tất cả các thành phần khác của PBS giao tiếp với module quản lý công việc qua một địa chỉ IP duy nhất. Chức năng của module quản lý công việc là cung cấp các dịch vụ, như là nhận/tạo ra các công việc lô, thay đổi các công việc, bảo vệ công việc khi có sự cố hệ thống, và thực hiện công việc. Module này quản lý một hoặc nhiều hàng đợi công việc, một công việc phải thuộc vào một hàng đợi. Các hàng đợi được server quản lý bởi một tập thuộc tính như kiểu, tài nguyên, tên,...

2. Module thực hiện công việc (PBS Mom): được chia nhỏ thành các thành phần chức năng sau: *Job Executor (JE)*, là tiến trình ngầm chịu trách nhiệm thực thi công việc. Khi thực hiện, JE nhận một bản copy công việc từ Server, xử lý công việc, và sau đó có thể sẽ đảm nhận luôn công việc trả kết quả về cho user nếu được Server yêu cầu. Mỗi một tiến trình ngầm JE chạy trên một máy tính trạm trong mạng; *Resource Monitor (RM)*, là bộ phận chịu trách nhiệm theo dõi, kiểm tra các nguồn tài nguyên của hệ thống và báo cho bộ lập lịch Scheduler.

3. Mô-đun quản lý tài nguyên (Job Scheduler): là thành phần chịu trách nhiệm xác định xem công việc nào sẽ chạy và chạy trên tài nguyên nào.



Hình 3-6 Cơ chế hoạt động của PBS

Tương tác giữa các module trên có thể được mô tả như sau:

1. Sự kiện kích hoạt module quản lý công việc bắt đầu một vòng lập lịch.
2. Module quản lý công việc gửi lệnh lập lịch cho bộ lập lịch.
3. Module lập lịch yêu cầu thông tin về tài nguyên từ module quản lý tài nguyên.
4. Module quản lý tài nguyên trả về các thông tin yêu cầu.
5. Bộ lập lịch yêu cầu các thông tin từ module quản lý công việc.
6. Module quản lý công việc gửi thông tin trạng thái của các công việc tới bộ lập lịch, bộ lập lịch đưa ra chính sách quyết định việc thực hiện công việc.
7. Bộ lập lịch gửi yêu cầu thực hiện công việc đến module quản lý công việc.

Module quản lý công việc gửi công việc đến các module thực thi công việc để thực hiện.

- ***Đệ trình công việc trong PBS***

Người sử dụng có thể thực hiện lệnh trực tiếp trên màn hình console để truyền công việc cũng như các yêu cầu của mình tới Server. Các shell [14] chuẩn của linux có thể giúp ta biên dịch mã lệnh để thực hiện các job.

Lệnh **qsub** là lệnh dùng để đệ trình công việc mình muốn thực hiện lên server. Cấu trúc lệnh như sau:

```
% qsub  [option]   mysubrun
```

trong đó **option** là các lựa chọn khi thực hiện công việc và **mysubrun** là công việc cần thực hiện. Các tuỳ chọn dùng để mô tả tài nguyên; trong ví dụ sau:

```
% qsub   -l  ncpus=16, walltime=4:0:0    mysubrun
```

ncpus là số CPU sử dụng để thực hiện công việc

walltime là thời gian dự kiến thực hiện công việc

Các công việc cũng có thể được miêu tả trong một file đặc tả, trong file thực hiện mô tả tất cả các thông số tài nguyên cần sử dụng, đầu vào, đầu ra… Khi đó việc đệ trình công việc trong PBS sẽ được thực hiện bằng một lệnh đơn giản như sau:

```
% qsub  mysubrun
```

Trong đó *mysubrun* là file mô tả công việc. Như vậy để có thể đưa một công việc từ trên lưới xuống cho PBS_Server thì ta phải làm thông suốt quá trình chuyển đổi công việc từ file đặc tả RSL của Globus sang file miêu tả công việc của PBS, vấn đề này sẽ được đề cập trong phần tiếp theo của đồ án.

3.3.2.3 Các yêu cầu đối với thành phần kết nối

Coi Cluster là một tài nguyên lưới như các tài nguyên khác, nó cung cấp một giao diện truy cập thống nhất cho các thành viên lưới khác sử dụng. Globus toolkit cung cấp các modul cho phép các thành phần của globus giao tiếp được với hầu hết các hệ quản lý cluster phổ biến như PBS, LSF, Condor..(Chương 2).

Như vậy, để một cluster trở thành một tài nguyên lưới ta sẽ cài Globus Toolkit lên nút chủ của cluster, thêm một mô-đun thực hiện giao tiếp với nút chủ của cluster và thực hiện các cấu hình cần thiết để cho Globus có thể giao tiếp được với thành phần quản lý tài nguyên địa phương của cluster.

Thông thường ở trên lưới luôn tồn tại hai loại công việc có sự khác nhau rõ rệt. Thứ nhất, là những công việc thông thường được thực hiện qua dịch vụ lưới, người sử dụng chỉ việc đưa ra một bộ dữ liệu đầu vào qua GridPortal và nhận kết quả trả về. Loại công việc thứ hai, là những công việc có tính chuyên môn cao, đòi hỏi khối lượng tính toán lớn, đối với những công việc này thì người sử dụng phải biết rõ về tài nguyên sẽ thực hiện công việc (siêu máy tính hay Cluster), viết chương trình thực hiện đối với loại tài nguyên tương ứng, sau đó đệ trình công việc thông qua giao diện GridPortal, lưới sẽ tìm ra tài nguyên tương ứng dựa trên thời gian thực hiện và số nút tính toán, lưới thực hiện truyền công việc cho cluster cùng với các tham số để thực hiện, cluster thực hiện công việc và sau đó trả lại kết quả cho lưới sau đó lưới mới thực hiện trả kết quả về cho người dùng. Việc kết nối giữa lưới dựa trên GT và cluster dựa trên PBS thực chất là thực hiện giao tiếp giữa thành phần GRAM trong lưới và thành phần PBS Server trên nút chủ của Cluster thông qua thành phần Globus-scheduler-pbs.

GRAM: cung cấp một giao diện chuẩn cho việc yêu cầu và sử dụng tài nguyên hệ thống từ xa để thực hiện các công việc. Đối với các công việc chỉ yêu cầu máy tính đơn hay siêu máy tính thì GRAM thực hiện công việc ngay trên máy tính đó. Đối với những công việc yêu cầu thực hiện trên Cluster thì GRAM sẽ chuyển nội dung công việc cho thành phần Globus-scheduler-pbs.

Globus-scheduler-pbs: thành phần này có nhiệm vụ chuyển nội dung công việc sang dạng mà PBS_Server có thể hiểu được.

PBS_Server: là nút chủ thực hiện lập lịch (địa phương), quản lý và thực hiện phân chia công việc thực hiện trên các nút con.

Như vậy có thể thấy việc kết nối cluster vào lưới không có nghĩa là chúng ta phải quản lý đến từng máy đơn trong cluster mà quản lý cluster một cách tổng thể. Lưới sẽ quản lý cả cụm máy tính trong cluster như một tài nguyên đơn nhất thông qua bộ quản lý tài nguyên địa phương PBS.

- ***Thành phần Globus Scheduler PBS***

Globus Toolkit có cung cấp cho chúng ta một số dịch vụ để giao tiếp với PBS_Server. Những dịch vụ này nằm trong gói cài đặt scheduler-pbs-3.2-src_bundle.tar.gz.

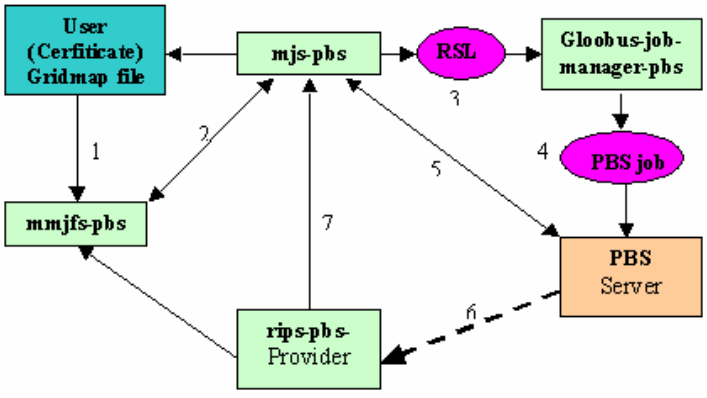
Sau khi cài đặt xong thì việc tiếp theo là cấu hình cho hệ thống. Đây thực sự là một vấn đề không đơn giản. Chi tiết về việc cấu hình và xây dựng một shell script cấu hình tự động sẽ được trình bày trong chương 4. Khi việc cấu hình hoàn tất, thực hiện khởi động trình chứa (globus-start-container) thì trong các dịch vụ của Globus Toolkit sẽ có thêm hai dịch vụ:

```
http://hostname:8080/ogsa/services/base/gram/PbsManagedJobFac
toryService
http://hostname:8080/ogsa/services/base/gram/MasterPbsManaged
JobFactoryService
```

Dịch vụ MasterPbsManagedJobFactoryService đóng vai trò là một giao diện đệ trình công việc, trực tiếp nhận yêu cầu công việc từ phía client. Đối với từng công việc dịch vụ MasterPbsManagedJobFactoryService sẽ gọi đến dịch vụ PbsManagedJobFactoryService sẽ sinh ra các thể hiện (instance) tương ứng trực tiếp làm việc với client.

- ***Hoạt động của Globus Scheduler Pbs***

Như đã trình bày ở trên, Globus Scheduler Pbs gồm 4 thành phần (mjs-pbs, mmjfs-pbs, rips-pbs-provider, globus-job-manager-pbs) giúp cho việc hoạt động đệ trình công việc lên PBS_Server thực hiện được. Giao tiếp giữa các thành phần này được mô tả trong hình sau:



Hình 3-7 Hoạt động của Globus Scheduler Pbs

1. Người dùng có giấy chứng nhận được cấp bởi nhà thẩm quyền, được thực hiện trên tài nguyên lưới (Quy định trong Grid-mapfile) thực hiện đệ trình công việc với trình quản lý công việc MMJFS-PBS.

2. Trình quản lý công việc MMJFS-PBS thực hiện sinh ra một thể hiện quản lý công việc phục vụ cho người dùng MJS-PBS.

3. Trình quản lý công việc MJS-PBS thực hiện lấy file đặc tả công việc của người dùng và chuyển cho thành phần biên dịch Globus-job-manager-pbs.

4. Thành phần Globus-job-manager-pbs thực hiện chuyển file đặc tả người dùng thành công việc cho PBS_Server.

5. Trình quản lý công việc người dùng yêu cầu PBS_Server thực hiện công việc.

6. PBS_Server thực hiện công việc và đưa kết quả trả về cho thành phần quản lý thông tin tài nguyên RIPS-PBS.

7. RIPS-PBS trả kết quả về cho trình quản lý công việc người dùng.

8. Trình quản lý công việc người dùng MJS-PBS trả kết quả về cho người sử dụng.

3.4 Kết chương

Chương này đã đưa ra các khái niệm, khía cạnh kỹ thuật và cơ chế triển khai, kết nối một số dạng lưới đang được triển khai tại các trung tâm nghiên cứu về tính toán lưới ở nước ta. Phần kiến thức được trình bày trong chương này sẽ làm tiền đề kết hợp với Chương 1 và Chương 2 để tiến hành xây dựng một lưới cụ thể trong chương tiếp theo.

CHƯƠNG 4. TRIỂN KHAI THỬ NGHIỆM

4.1 Lập bản thiết kế kiến trúc lưới

Việc xây dựng một hệ thống tính toán lưới không chỉ đơn thuần là cài đặt các phần mềm, các ứng dụng trên các tài nguyên kết nối sẵn có mà phải tuân theo quy trình quy hoạch, thiết kế một cách bài bản, chuyên nghiệp. Về cơ bản, việc thiết kế lưới bao gồm một số bước chính sau [2]:

- Khảo sát yêu cầu nghiệp vụ của tổ chức
- Quy hoạch hạ tầng mạng sẵn có
- Lựa chọn loại lưới
- Lựa chọn topo lưới
- Lựa chọn hạ tầng (hạ tầng phần mềm, phần cứng, các chuẩn mở, cơ chế an toàn, bảo mật cho lưới…)
- Chính sách quản trị lưới..
- Triển khai một lưới

Ngoài ra, khi thiết kế một lưới thì kiến trúc lưới đó cần phải thoả mãn được các yêu cầu cơ bản như: tính an toàn, tĩnh sẵn sàng và hiệu suất cao.

➢ ***Khảo sát yêu cầu***

Trong thiết kế kiến trúc lưới, bước khảo sát yêu cầu là bước rất quan trọng, kết qủa của nó sẽ ảnh hưởng trực tiếp đến tất cả các bước tiếp theo. Thường trong bước này, người thiết kế phải đưa ra được những thông tin cần thiết như mục đích của việc xây dựng lưới là gì? Lưới có những yêu cầu gì về nghiệp vụ, hạ tầng và ứng dụng? Cuối cùng cần phải tinh chỉnh và xem xét lại xem những yêu cầu này có phù hợp với hạ tầng phần cứng và phần mềm cũng như nguồn nhân lực sẵn có hay không.

Sau khi đã xác định đươc các yêu cầu đặt ra cho lưới, đồng thời nhận thấy rằng hạ tầng hiện có là đủ đáp ứng để xây dựng lưới, ta phải tiến hành qui hoạch lại hạ tầng mạng để phục vụ việc triển khai lưới được thuận tiện và khoa học.

➢ ***Lựa chọn loại lưới***

Thường thì tuỳ theo yêu cầu nghiệp vụ của lưới mà ta sẽ chọn ra loại lưới thích hợp. Nhưng nhìn chung, các lưới được triển khai thường rơi vào hai dạng sau:

Lưới tính toán (Computational Grid): dùng khi muốn kết hợp và tăng sức mạnh tính toán của các hệ thống tính toán phân tán. Đặc trưng cơ bản nhất của lưới tính toán là dựa trên sự tích hợp các đơn vị có sức mạnh tính toán nhưng khả năng lưu trữ không cao. Lợi ích của các lưới tính toán là giảm thiểu chi phí sở hữu và rút ngắn thời gian phát triển. Nó được dùng cho mục đích nâng cao hiệu suất sử dụng các hệ thống máy tính và hệ thống các ứng dụng sẵn có và dùng để giải các bài toán khoa học, các bài toán mô phỏng, các bài toán đánh giá cơ hội đầu …

Lưới dữ liệu (Data Grid): chủ yếu dùng vào việc cung cấp khả năng truy cập đến các nguồn dữ liệu hỗn hợp, phân tán và bảo mật cho các thực thể tham gia lưới. Lưới dữ liệu có thể được hiểu như các cơ sở dữ liệu liên hợp, nó giữ vai trò liên kết các dữ liệu rời rạc thành một cơ sở dữ liệu ảo thống nhất và người dùng có thể truy cập cơ sở dữ liệu này qua một giao diện duy nhất. Lưới dữ liệu thích hợp với các tổ chức, các doanh nghiệp có nhu cầu mở rộng khả năng khai phá, sử dụng dữ liệu trên phạm vi rộng nhằm tối ưu hoá việc sử dụng hạ tầng thiết bị lưu trữ sẵn có và giảm thiểu hoá sự phức tạp trong việc quản lý dữ liệu.

- ***Lựa chọn topo lưới***

Hiện nay khi triển khai một lưới, người thiết kế thường lựa chọn một trong số các topo lưới thông dụng sau [2]:

Intragrid: thường được triển khai trong các tổ chức hoặc doanh nghiệp. Dựa trên kiến trúc mạng LAN hoặc mạng Intranet dùng riêng của tổ chức, doanh nghiệp đó. Nó có: Băng thông và mức độ sẵn sàng cao; Cơ chế an toàn, bảo mật riêng rẽ; Môi trường tác nghiệp độc lập. Topo lưới dạng này rất phù hợp khi triển khai các lưới tính toán hoặc lưới dữ liệu.

Extragrid: được thiết lập dựa trên hai hoặc nhiều Intragrid. Đặc trưng của loại topo này là: Cơ chế an toàn, bảo mật phân tán; Có sự tham gia của nhiều tổ chức doanh nghiệp; Dựa trên mạng WAN. Extragrid thích hợp với các tổ chức

muốn xây dựng kết nối mạng với các đối tác của mình (B2B) nhằm chia sẻ tài nguyên, dữ liệu dựa trên sự tin tưởng lẫn nhau.

Intergrid: xây dựng trên mạng WAN hoặc Internet và được sử dụng bởi các công ty công nghệ, tập đoàn công nghiệp, hoặc nhà sản xuất công nghiệp. Đặc trưng của Intergrid là: Có sự tham gia của nhiều tổ chức; Kết nối nhiều đối tác; Kết nối nhiều mạng liên kết; Cơ chế an ninh phức tạp, phân tán.

- ***Lựa chọn hạ tầng lưới***

Khái niệm hạ tầng ở đây bao gồm cả phần cứng vật lý và các phần mềm được sử dụng để kết nối các máy tính của một lưới lại với nhau. Hạ tầng này cung cấp các dịch vụ cho việc kết nối, đảm bảo an toàn và quản trị.

Hạ tầng bảo mật: giải pháp bảo mật hay được áp dụng cho các hệ thống tính toán lưới là sử dụng tường lửa. Giải pháp này bảo vệ được các máy chủ trong lưới tránh khỏi những tấn công từ bên ngoài và tạo lập thêm một hàng rào bảo vệ, ngăn cản những truy nhập không mong muốn từ người sử dụng bên ngoài lẫn bên trong.

Hạ tầng mạng: phải cung cấp băng thông đủ lớn cho các ứng dụng và dịch vụ trên nó. Hạ tầng mạng cần phải thường xuyên được bảo trì, tuỳ chỉnh nhằm đảm bảo hiệu suất hoạt động và tính sẵn sàng cao nhất.

Hạ tầng lưu trữ: đây là một trong những sức mạnh vô tận của hệ thống tính toán lưới. Khi thiết kế hệ thống, người thiết kế phải trả lời được các câu hỏi như: cách thức nào dùng để bảo đảm an toàn cho các thiết bị lưu trữ, cách thức nào dùng trong nhân bản dữ liệu, và làm sao để quản lý các thiết bị này một cách hiệu quả nhất. Mục tiêu cuối cùng của việc thiết kế các lưới là phải đảm bảo sự sẵn sàng của dữ liệu cho các tài nguyên và người dùng khi có yêu cầu.

- ***Lựa chọn tập các dịch vụ cơ bản cho lưới***

Tập các dịch vụ này gồm các thành phần: Bộ cung cấp các tính năng bảo mật; Tập các thành phần cơ bản cuả GT như Grid clients; CA; GateKeepers; MDS (GRIS/GIIS); Các chứng chỉ số cho việc xác thực; Bộ lập lịch công việc cơ bản; Các thiết bị hạ tầng thông dụng như mạng quản trị, tường lửa, hệ thống phát hiện truy cập trái phép; Mạng các thiết bị lưu trữ …

- ***Quản trị lưới tính toán***

Đối với người quản trị lưới, cần phải nắm rõ các sản phẩm liên quan và các tính năng của hệ thống tính toán lưới. Các sản phẩn liên quan đến hệ thống tính toán lưới bao gồm: Phần mềm tầng trung gian; Hệ thống giám sát hiệu suất lưới; Grid portals; Môi trường lập trình; Bộ lập lịch; Hệ thống phát triển và kiểm thử lưới.

Các tính năng của hệ thống tính toán lưới: Quản lý dịch vụ; Truyền thông dịch vụ; Quản lý chính sách; Lựa chọn và triển khai các phần mềm trung gian, lựa chọn các ứng dụng, trả kết quả về cho các ứng dụng; Điều khiển dịch vụ; SLAs; Quản lý hiệu suất/tài khoản/lỗi; an toàn và bảo mật.

- ***Triển khai một lưới***

Một lưới tính toán đơn giản có thể được xây dựng dựa trên một thư viện lập trình hỗ trợ các tính năng tính toán lưới, như sử dụng các tính năng được cung cấp bởi bộ công cụ Globus Toolkit. Tuy nhiên, khi hệ thống được mở rộng và trở nên phức tạp, ta cần phải tiếp cận theo phương thức khác.

Đầu tiên là bước thiếp lập hạ tầng cho một lưới tính toán đơn giản dựa trên Globus Toolkit bao gồm các cài đặt trên máy chủ và các máy trạm như: Hệ điều hành, các thư viện, bộ công cụ GT; Thiết lập các dịch vụ, đơn vị chứng thực, bộ quản lý tài nguyên …

Tiếp theo là các bước thiết lập ứng dụng, dịch vụ cho hệ thống tính toán lưới vừa được cài đặt dựa trên Globus Toolkit hoặc bất cứ bộ công cụ nào dựa trên chuẩn OGSA bao gồm: Định nghĩa các dịch vụ của lưới bằng ngôn ngữ WSDL; Sinh mã Java cho việc định nghĩa WSDL; Viết mã cài đặt phía máy chủ, máy khách; Triển khai và thử nghiệm các dịch vụ lưới sử dụng trình duyệt OGSA.

Sau đây là danh sách công việc cần thực hiện: Dự kiến và quy định tài nguyên của lưới; Lựa chọn phần mềm trung gian; Lựa chọn các công cụ quản trị, xây dựng quy trình quản trị đối với phần mềm trung gian; Cài đặt, thử nghiệm, tuỳ chỉnh và phát hành các phần mềm, ứng dụng của lưới; Thiết lập bộ điều khiển dịch vụ, các công cụ, quy trình kiểm soát, điều khiển an ninh; Thiết lập các công cụ quản lý lỗi, hiệu năng, hạ tầng mang tính tích hợp.

➢ ***Các yêu cầu cơ bản của lưới sau khi thiết lập***

Tính an toàn: lưới sau khi triển khai phải giảm thiểu rủi ro đến mức chấp nhận được, nghĩa là trong bản thiết kế cần phải vạch rõ các yêu cầu về an toàn, bảo mật của hệ thống, đồng thời phải đưa ra các công cụ và các biện pháp nhằn tăng tính an toàn và bảo mật cho hệ thống. Mức độ an toàn và bảo mật của hệ thống tính toán lưới còn phụ thuộc vào topo lưới và dữ liệu cũng như tài nguyên cần bảo mật. Mức độ bảo mật cũng khác nhau giữa các lưới được dùng trong các công ty có tính chất nghiệp vụ khác nhau. Bản thân các mô hình an toàn và bảo mật cho một hệ thống tính toán lưới cũng có những rủi ro tiềm ẩn. Do đó, khi thiết kế một mô hình bảo mật cho lưới cũng cần tính đến một vài phương án dự phòng. Ngoài ra, nên có tường lửa, hệ thống phát hiện truy cập trái phép, các giải pháp phòng chống virus,…

Tính sẵn sàng: được hiểu một cách đơn giản là phần trăm thời gian mà hệ thống đáp ứng trở lại các yêu cầu của thực thể tham gia lưới. Trong giai đoạn thiết kế, cũng cần đưa ra mức độ sẵn sàng của hệ thống. Dựa trên mức độ này, người thiết kế xác định được tài nguyên dư thừa cần có của hệ thống và dự phòng được các phương thức xử lý sự cố hỏng hóc khi cần thiết.

Cuối cùng, dựa trên những kiến thức thu được từ việc nghiên cứu về tính toán lưới và các công cụ liên quan đã được trình bày trong các chương trước. Đồng thời với sự giúp đỡ của thầy hướng dẫn và các bạn trên trung tâm tính toán hiệu năng cao. Tôi và các bạn trên trung tâm đã tiến hành cài đặt thử nghiệm một lưới với mục đích hiểu sâu hơn các vấn đề đã nghiên cứu đồng thời tạo ra một lưới dùng để phục vụ việc thử nghiệm các nghiên cứu tính toán lưới sau này.

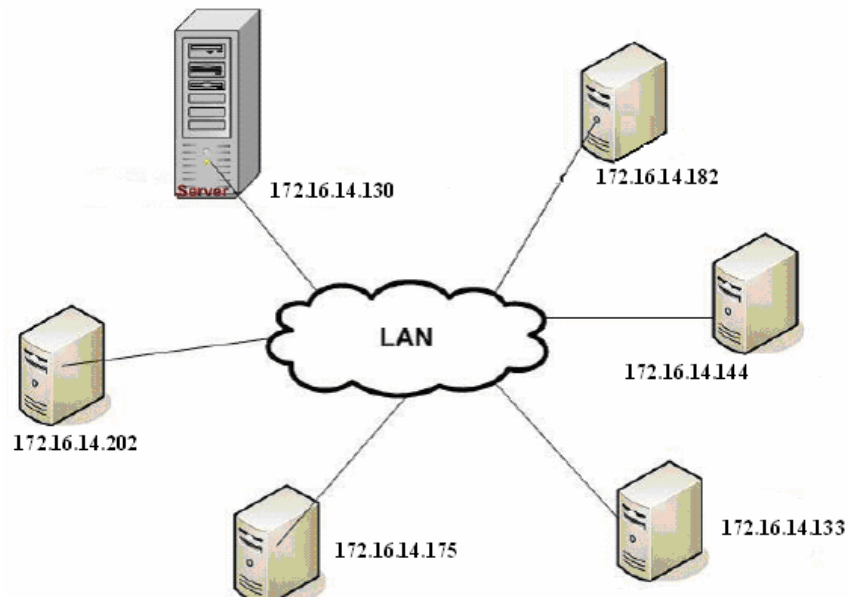
4.2 Cài đặt một Grid

Phần này liệt kê các bước và giải thích ngắn gọn quá trình cài đặt hệ thống cơ sở hạ tầng cho một lưới dựa trên bộ Globus Toolkit 4.

4.2.1 Cấu hình phần cứng của lưới

Topo lưới trên trung tâm được bố trí như hình 4.1, hệ thống bao gồm: Một nút chính là Gridsv-172.16.14.130 và Năm nút tính toán là: Alpha-172.16.14.182;

Beta-172.16.14.144; Gama-172.16.14.133; Delta-172.16.14.175; Omega-172.16.14.202. Tại thời điểm cài đặt thử nghiệm tất cả các máy tính này đều là PenIII800, và được kết nối với nhau qua mạng LAN.



Hình 4-1 Mô hình lưới được triển khai thử nghiệm

4.2.2 Yêu cầu trước khi cài đặt

Các nút lưới là các máy tính PC Intel Pentium đã nối mạng với nhau qua Fast Ethernet, đã cài Linux Fedora Core 3 và tất cả các gói phần mềm cần thiết [5] cho việc cài đặt Globus Toolkit.

Để các nút trong mạng có thể nhìn và liên lạc được với nhau thì file */etc/hosts* trên tất cả các nút lưới phải có cùng nội dung và chứa đầy đủ 6 địa chỉ IP cũng như tên của các máy trực tiếp tham gia lưới.

4.2.3 Cài đặt cho nút chính

Trước khi thực hiện cài đặt cho nút chính, ta tạo một user "*globus*", đây là user thực hiện công việc quản trị Globus Toolkit. Phải đăng nhập vào user *root* (tạo khi cài đặt Fedora Core 3) để thực hiện lệnh tạo user, vì chỉ *root* mới có quyền ghi vào các file hệ thống.

```
[root]#adduser -p "globus" -g users -d /home/globus globus
[root]#mkdir /usr/local/GlobusToolkit4.0.1
[root]#ln -s /usr/local/GlobusToolkit4.0.1 /usr/local/globus
[root]#chown globus:users /usr/local/GlobusToolkit4.0.1
```

Bước 1: Cài gói gt4.0.1-x86_fc_3-binary-installer bằng user *globus* vừa tạo:

```
[globus]$tar xzf /opt/gt4.0.1-x86_fc_3-binary-
installer.tar.gz
[globus]$export JAVA_HOME=/usr/local/java
[globus]$export ANT_HOME=/usr/local/ant
[globus]$export GLOBUS_LOCATION=/usr/local/globus
[globus]$cd gt4.0.1-x86_fc_3-binary-installer
[globus]$./configure --prefix=/usr/local/globus --with-
             iodbc=/usr/local/lib
[globus]$make
[globus]$make install
```

Bước 2: Cấu hình an toàn bảo mật cho nút chính bằng user *globus*. Đây là bước tạo simpleCA, ký chứng thực cho host:

```
[globus]$$GLOBUS_LOCATION/setup/globus/setup-simple-ca
```

lựa chọn các tuỳ chọn cho lệnh này như trong phụ lục A của [5].

Login vào user *root,* gõ lệnh:

```
[root]#$GLOBUS_LOCATION/setup/globus_simple_ca_ebb88ce5_setup
/setup-gsi
[root]#source $GLOBUS_LOCATION/etc/globus-user-env.sh
[root]#grid-cert-request -host `hostname`
```

Login vào user *globus*, gõ lệnh:

```
[globus]$grid-ca-sign -in /etc/grid-
security/hostcert_request.pem -out hostsigned.pem
```

Login vào user *root*, gõ lệnh:

```
[root]#cp ~globus/hostsigned.pem /etc/grid-
security/hostcert.pem
[root]#cp /etc/grid-security/hostcert.pem/etc/grid-
security/containercert.pem
[root]#cp /etc/grid-security/hostkey.pem/etc/grid-
security/containerkey.pem
```

```
[root]#chown globus:users container*.pem
```

Cấp và ký chứng thực người dùng "*usergrid*". Tạo người dùng có tên là *usergrid*. Đây là người dùng sẽ sử dụng các dịch vụ của hệ thống Grid. Dùng lệnh tạo user như với user *globus*, sau đó tạo tạo simpleCA, ký chứng thực cho host như bước 2.

Các bước tiếp theo là cài đặt và cấu hình các dịch vụ, lệnh chi tiết trong [5].

Bước 3: Cài đặt dịch vụ **GRAM**

Login vào user root, gõ lệnh cấu hình sudo:

```
[root]#visudo
```

sau đó, thêm 2 dòng sau vào file *etc/sudoers*:

```
# Globus GRAM entries
Runas_Alias GRAM_USERS = usergrid
globus  ALL=(GRAM_USERS)
NOPASSWD: /usr/local/globus/libexec/globus-gridmap-and-
execute -g  /etc/grid-security/grid-mapfile
/usr/local/globus/libexec/globus-job-manager-script.pl *
globus  ALL=(GRAM_USERS)
NOPASSWD: /usr/local/globus/libexec/globus-gridmap-and-
execute  -g /etc/grid-security/grid-mapfile
/usr/local/globus/libexec/globus-gram-local-proxy-tool *
```

Bước 4: Cài đặt **GridFTP**

Login vào user *root*, dùng bất kỳ trình soạn thảo dạng console nào của Fedora để tạo file */etc/xinetd.d/gridftp* có nội dung như sau:

```
service gsiftp
{
      instances  = 100
      socket_type      = stream
      wait       = no
      user       = root
      env        += GLOBUS_LOCATION=/usr/local/globus
      env        += LD_LIBRARY_PATH=/usr/local/globus/lib
      server     = /usr/local/globus/sbin/globus-gridftp-
server
```

```
        server_args     = -i
        log_on_success+= DURATION
        nice        = 10
        disable     = no
}
```

Sau đó gõ lệnh:

```
[root]#echo "gsiftp    2811/tcp" >> /etc/services
[root]#/etc/init.d/xinetd reload
[root]#netstat -an | grep 2811
```

Bước 5: Khởi động **GT container** tự động khi boot máy

Login vào user *globus*, tạo file *$GLOBUS_LOCATION/bin/start-stop* có nội dung như sau:

```
#!/bin/sh
set –e
export GLOBUS_LOCATION=/usr/local/globus
export JAVA_HOME=/usr/local/java
export ANT_HOME=/usr/local/ant
export GLOBUS_OPTIONS="-Xms256M -Xmx512M"
. $GLOBUS_LOCATION/etc/globus-user-env.sh
cd $GLOBUS_LOCATION
case "$1" in
       start)
          $GLOBUS_LOCATION/sbin/globus-start-container-
detached -p 8443
      ;;
       stop)
          $GLOBUS_LOCATION/sbin/globus-stop-container-
detached
      ;;
       *)
          echo "Usage: globus {start|stop}" >&2
          exit 1
      ;;
esac
```

```
exit 0
[globus]$chmod +x $GLOBUS_LOCATION/bin/start-stop
```

Login vào user *root*, tạo file */etc/init.d/globus* có nội dung sau:

```
#!/bin/sh –e
## BEGIN chkconfig header
# chkconfig: 345 80 30
# processname: /usr/local/globus/bin/start-stop
# description: sysv init script to start GT container
## END chkconfig header
case "$1" in
       start)
          su - globus /usr/local/globus/bin/start-stop start
      ;;
       stop)
          su - globus /usr/local/globus/bin/start-stop stop
      ;;
       restart)
          $0 stop
          sleep 1
          $0 start
      ;;
       *)
          echo "Usage: $0 {start|stop|restart}n" >&2
      ;;
esac
exit 0
[root]#755 /etc/init.d/globus
[root]#chkconfig --add globus
[root]#service globus start
```

Bước 6: Cấu hình **RFT**. Cấu hình hệ quản trị cơ sở dữ liệu PostgresSQL

Thêm dòng "*host rftDatabase "globus" "172.16.14.130" 255.255.255.0 trust*" vào tập tin */var/lib/pgsql/data/pg_hba.conf*, trong đó 172.16.14.130 là địa chỉ IP của nút ta đang cài đặt.

```
[root]#/etc/init.d/postgresql restart
```

```
[root]#su postgres -c "createuser -P globus"
```

Tạo cơ sở dữ liệu ban đầu cho RFT

```
[globus]$createdb rftDatabase
[globus]$psql -d rftDatabase -f
[globus]$$GLOBUS_LOCATION/share/globus_wsrf_rft/rft_schema.sq
l
```

Thay đổi mật khẩu trong file

$GLOBUS_LOCATION/etc/globus_wsrf_rft/jndi-config.xml thành "globus"

4.2.4 Cài đặt các nút tính toán

Phần này trình bày các bước tiếp theo để cài đặt cơ sở hạ tầng Grid lên các nút tính toán. Cài đặt này phải thực hiện lặp trên tất cả các nút tính toán có trong lưới. Trên các nút tính toán việc cài đặt như tạo user hay cài đặt **GT4**, **GridFTP**, **GT container**, **GRAM** và **RFT** đều thực hiện tương tự như với nút server. Riêng cài đặt **GSI** thì có chút khác biệt sau:

Dùng user *globus*, sao chép file *globus_simple_ca_ebb88ce5_setup-0.18.tar.gz* trong thư mục ~globus/.globus/simpleCA trên nút chính vào thư mục home của user *globus* trên nút tính toán.

```
[globus]$export GLOBUS_LOCATION=/usr/local/globus
[globus]$$GLOBUS_LOCATION/sbin/gpt-build
 globus_simple_ca_ebb88ce5_setup-0.18.tar.gz
[globus]$$GLOBUS_LOCATION/sbin/gpt-postinstall
```

Login vào user *root*, gõ lệnh:

```
[root]#export GLOBUS_LOCATION=/usr/local/globus
[root]#source $GLOBUS_LOCATION/etc/globus-user-env.sh
[root]#$GLOBUS_LOCATION/setup/globus_simple_ca_ebb88ce5_setup
/setup-gsi
```

Tạo yêu cầu cấp chứng thực và ký chứng thực host cho nút tính toán bằng user *root*:

```
[root]#grid-cert-request -host `hostname`
```

copy tập tin */etc/grid-security/hostcert_request.pem* tới người quản trị globus trên nút CA: globus@gridsv.it.bkvn vào thư mục *~globus/.globus/hosts/<hostname>*, trong đó (<hostname>: tên của nút tính toán xin cấp chứng thực)

Người quản trị globus@gridsv.it.bk.vn sẽ ký chứng thực bằng lệnh

```
grid-ca-sign -in.globus/hosts/<hostname>/hostcert_request.pem
-out hostcert.pem
```

sau đó chuyển tập tin *hostcert.pem* này vào lại thư mục */etc/grid-security* trên nút tính toán bằng quyền *root*.

```
[root]#cp ~globus/hostsigned.pem /etc/grid-
security/hostcert.pem
[root]#cp /etc/grid-security/hostcert.pem/etc/grid-
security/containercert.pem
[root]#cp /etc/grid-security/hostkey.pem/etc/grid-
security/containerkey.pem
[root]#chown globus:users container*.pem
[root]#ls -l /etc/grid-security/*.pem
[root]#echo ""/O=Grid/OU=GlobusTest/OU=simpleCA-
gridca.math.hcmuns.edu.vn/OU=math.hcmuns.edu.vn/CN=igrid"
igrid" >> /etc/grid-security/grid-mapfile
```

Tạo yêu cầu cấp chứng thực và ký chứng thực người dùng *usergrid* trên nút tính toán. Bước này chỉ cần thực hiện nếu cho phép người dùng *usergrid* login vào nút tính toán để sử dụng lưới.

4.2.5 Đồng bộ thời gian giữa các nút trong lưới

Đây là bước khá quan trọng bởi các chứng thực đều chỉ có hiệu lực trong một khoảng thời gian. Yêu cầu tất cả các nút đều phảicó gói *ntp-4.0.0* trở.

- ***Cài đặt NTP server trên nút chính***

Đầu tiên thay đổi nội dung tập tin */etc/ntp.conf* thành:

```
restrict default noserve
restrict 127.0.0.1
restrict 172.16.14.0 mask 255.255.255.0
server  0.pool.ntp.org
```

```
server  1.pool.ntp.org
server  2.pool.ntp.org
fudge   172.16.14.0 stratum 10
driftfile /etc/ntp/drift
broadcastdelay  0.008
authenticate no
```

Cập nhật ngày giờ cho nút chính:

```
[root]#ntpdate -u 0.pool.ntp.org
```

(lặp lại lệnh này cho đến khi giờ của nút chính là chính xác)

Khởi động NTP server:

```
[root]#chkconfig ntpd on
[root]#/etc/init.d/ntpd
```

Cập nhật trạng thái cho NTP server:

```
[root]#ptrace 0.pool.ntp.org
[root]#ntptrace 1.pool.ntp.org
[root]#ntptrace 2.pool.ntp.org
[root]#ntpq –p
[root]#ntptrace localhost
```

- ***Cài đặt NTP client trên các nút tính toán***

```
[root]#ntpdate -b 172.16.14.130
```

(Chạy nhiều lần lệnh này cho đến khi thời gian của hệ thống là chính xác)

```
[root]#chkconfig crond on
```

Thêm tập tin */etc/cron.hourly/ntp.client* nội dung như sau:

```
#!/bin/sh
# Update system time every hour from time server:
172.29.3.131 ntpdate -b 172.16.14.130
[root]#chmod +x /etc/cron.hourly/ntp.client
[root]#/etc/init.d/crond reload
```

4.2.6 Cấu hình các dịch vụ mức lưới

Sau khi đã cài đặt xong hết các gói cần thiết, ta tiến hành cấu hình các dịch vụ IndexService, WebMDS, MyProxy server. Thông tin và lệnh cấu hình được thực hiện như trong [5].

- ***Cấu hình IndexService***

Ta sẽ cấu hình để các nút tính toán đăng ký dịch vụ **Index** với nút chính 172.29.3.130, bằng cách thêm dòng:

```
<upstream>https://gridca.math.hcmuns.edu.vn:8443/wsrf/service
s/DefaultIndexService</upstream>
```

vào tập tin *$GLOBUS_LOCATION/etc/globus_wsrf_mds_index/hierarchy.xml*, sau đó khởi động lại GT container:

```
[globus]$globus-stop-container-detached
[globus]$sleep 5
[globus]$globus-start-container-detached
```

- ***Cấu hình WebMDS***

Đây là ứng dụng Web cho phép hiển thị các thông tin MDS trong một lưới. Chỉ cài đặt WebMDS trên nút chính bằng gói **Tomcat** *jakarta-tomcat-4.1.31.tar.gz*:

```
[root]#tar zxf /opt/jakarta-tomcat-4.1.31.tar.gz
[root]#ln -s jakarta-tomcat-4.1.31 tomcat
[root]#echo "export CATALINA_HOME=/usr/local/tomcat"
>>/etc/profile
[root]#echo "export PATH=$CATALINA_HOME/bin:$PATH" >>
/etc/profile
```

- ***Cấu hình và triển khai WebMDS vào Tomcat container***

Sửa tham số "endpoint" trong file

$GLOBUS_LOCATION/lib/webmds/conf/indexinfo thành:

```
"https://172.29.3.130:8443/wsrf/services/DefaultIndexService"
[root]#$GLOBUS_LOCATION/lib/webmds/bin/webmds-create-context-
file
$CATALINA_HOME/conf/Catalina/localhost
[root]#$CATALINA_HOME/bin/startup.sh
```

- ***Cài đặt và cấu hình MyProxy server trên nút chính***

Dùng để quản lý các chứng thực người dùng tại các nút khác trong lưới.

```
[root]#  cp $GLOBUS_LOCATION/etc/myproxy-server.config /etc
```

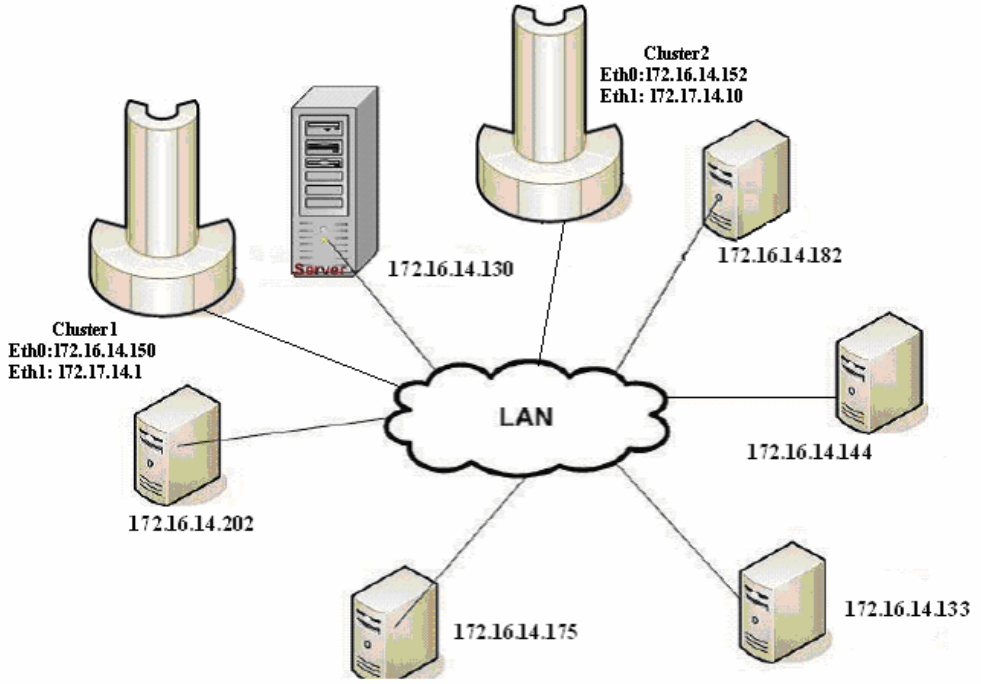
Sau đó, soạn lại tập tin */etc/myproxy-server.config* như [5].

4.3 Kết nối một Cluster vào Grid

Sau khi đã có một lưới, thực hiện thiết lập các mạng con, mỗi mạng con được coi là một hệ thống tính toán song song phân cụm (Cluster). Thực hiện kết nối các mạng con này vào lưới.

4.3.1 Cấu hình phần cứng

Trên trung tâm triển khai 2 cluster để kết nối vào hệ thống lưới đã được thiết lập trong phần trước với topo mạng như hình dưới:



Hình 4-2 Topo mạng triển khai kết nối Cluster vào lưới

Đợt cài đặt thử nghiệm này sử dụng hai cluster do một nhóm sinh viên thực tập trên trung tâm thiết lập trước đó, cluster thứ nhất gồm 4 máy tính có địa chỉ IP từ 172.17.14.1 đến 172.17.14.4 đã được cài đặt Globus Toolkits, GlobusPbsScheduler, PBS và MPICH. Sử dụng máy có địa chỉ 172.17.14.1 làm máy chủ, đồng thời nó có một địa chỉ IP dùng để giao tiếp với bên ngoài là 172.16.14.150. Cluster thứ hai gồm 6 máy tính được thiết lập địa chỉ IP từ

172.17.14.10 đến 172.17.14.15, trên các máy này đã cài đặt Globus Toolkits, GlobusPbsScheduler, PBS và MPICH. Ở cluster thứ hai, sử dụng máy có địa chỉ 172.17.14.10 làm máy chủ, máy chủ này có một địa chỉ IP dùng để giao tiếp với bên ngoài là 172.16.14.152.

Bên cạnh các cluster thêm mới, hệ thống lưới tính toán vẫn giữ nguyên toàn bộ các máy đơn đã thiết lập ở mục trước.

4.3.2 Cấu hình cluster-based PBS

Công việc cấu hình cluster-based PBS được thực hiện trên nút chủ của cluster. PBS đã cung cấp cho chúng ta môt lệnh quản lý cấu hình PBS dùng cho việc cấu hình trên giao diện dòng lệnh đó là lệnh **qmgr**.

- ***Các lệnh cấu hình PBS***

Các lệnh cấu hình cho PBS thường có cấu trúc như sau:

```
command server [names] [attr OP value[,...]]
command queue  [names] [attr OP value[,...]]
command node   [names] [attr OP value[,...]]
```

Ở đậy "command" thực hiện một tác động lên các đối tượng "*server*", "*queue*", "*node*". Các lệnh này bao gồm [14]:

1. Active: dùng để kích hoạt đối tượng
2. Create: tạo một đối tượng mới như hàng đợi hoặc nút
3. Delete: xóa một đối tượng hàng đợi hoặc nút
4. Help: hiển thị thông tin chi tiết của lệnh
5. List: liệt kê các thuộc tính và giá trị của đối tượng
6. Print: hiển thị các hàng đợi đã được thiết lập và các thuộc tính của server
7. Set: định nghĩa hoặc thêm các thuộc tính cho đối tượng, ngược lại Unset

- ***Cấu hình cho đối tượng Server***

Sử dụng các thuộc tính của lệnh cấu hình PBS [14] như:

1. scheduling: kích hoạt hoặc tắt bộ lập lịch
2. max_user_run: số công việc tối đa một người có thể thực hiện cùng lúc
3. acl_host_enable: cho phép server sử dụng danh dách máy trong acl_hosts

4. acl_hosts: danh sách các máy có thể yêu cầu từ server

5. default_queue: hàng đợi mặc định dùng khi ko được chỉ rõ tên

6. log_events: độ dài xâu kỹ tự sẽ ghi lại các sự kiện xảy ra trong hệ thống

7. mail_from: định danh của người dùng mà server sẽ dùng để gửi thư

8. queue_other_job: cho phép người dùng không sở hữu công việc nhưng xem được tình trạng của nó

9. resources_default: tập các giá trị mặc định

10. scheduler_interation: khoảng thời gian giữa các lần lập lịch

11. default_node: nút tính toán mặc định dùng khi không chỉ rõ tên nút

để cấu hình cho server.

Toàn bộ việc cấu hình cho đối tượng Server được thực hiện bằng các lệnh như trong shell script ở [17].

- ***Cấu hình đối tượng hàng đợi (queue)***

Trong PBS có hai kiểu hàng đợi khác nhau [14]: hàng đợi định tuyến (routing) và hàng đợi thực thi (execution). Một hàng đợi định tuyến dùng để chuyển công việc từ hàng đợi này sang một hàng đợi khác trong PBS Server. Còn hàng đợi thực thi dùng để thực hiện công việc, công việc tồn tại trong nó suốt quá trình chạy. Thông thường một PBS Server muốn hoạt động tốt phải có cả hai loại hàng đợi này.

Các thuộc tính của hàng đợi chia thành ba nhóm:

- Nhóm thuộc tính chỉ áp dụng với hàng đợi thực thi
- Nhóm thuộc tính chỉ áp dụng với hàng đợi định tuyến
- Nhóm thuộc tính áp dụng với cả hai kiểu hàng đợi

Các thuộc tính của hàng đợi định tuyến

1. queue_type_route: là kiểu hàng đợi định tuyến, dùng để phân công việc

2. max_running: số công việc tối đa có thể năm trong hàng đợi cùng lúc

3. route_destination: các hàng đợi mà công việc sẽ được chuyển đến

4. enable: cho phép nhận/không nhận công việc mới

5. started: công việc được lập lịch sẽ được thực thi từ hàng đợi này

Các thuộc tính của hàng đợi thực thi

1. queue_type_excecute: kiểu hàng đợi thực thi

2. priority: mức độ ưu tiên của hàng đợi này so với các hàng đợi .

3. max_running: số công việc tối đa đồng thời có trong hàng đợi

4. resource_max: lượng tài nguyên lớn nhất có thể được yêu cầu cho một công việc, ngược lại là resource_min

5. resource_default: tài nguyên mặc định

6. enable: cho phép nhận/không nhận công việc mới

7. started: công việc được lập lịch sẽ được thực thi từ hàng đợi này

Việc cấu hình đối tượng hàng đợi định tuyến và hàng đợi thực thi được thực hiện thông qua các lệnh trong shell script ở [17].

Để tạo ra các loại hàng đợi khác nhau ra chỉ cần thay đổi các giá trị nhập cho các thuộc tính của lệnh trong shell script dùng để tạo một hàng đợi thực thi có tên là small, mức độ ưu tiên 100, dùng để thực hiện các công việc có thời gian yêu cầu ít hơn 20 phút.

```
# Create and define queue small
Qmgr: create queue small
Qmgr: set queue small queue_type = Execution
Qmgr: set queue small Priority = 100
Qmgr: set queue small max_running = 10
Qmgr: set queue small resources_max.cput = 00:20:00
Qmgr: set queue small resources_default.cput = 00:20:00
Qmgr: set queue small enabled = True
Qmgr: set queue small started = True
```

Tất cả các shell script ở [17] đã được dùng và kiểm thử trên trung tâm tính toán hiệu năng cao, và cho kết quả khá tốt.

4.3.3 Cấu hình lưới dựa trên GT

Trước khi thực hiện cấu hình cho lưới ta cần phải thực hiện cài đặt Globus Toolkits 4.0.1 trên nút chủ của cluster, cài đặt gói phần mềm *scheduler-pbs-3.2-src_bundle.tar.gz*. Sau đây sẽ là một số bước cấu hình cần thiết để thiết lập PBS là trình quản lý công việc cho lưới.

```
% gpt-postinstall
```

lệnh này thực hiện khởi tạo các cấu hình mặc định cho lưới.

```
$GLOBUS_LOCATION/setup/globus/setup-mmjfs-pbs
```

lệnh này thực hiện cấu hình để thêm dịch vụ MasterPbsManagedJobFactoryService vào trong các dịch vụ của Globus Toolkit.

```
$GLOBUS_LOCATION/setup/globus/setup-mjs-pbs
```

lệnh này thực hiện cấu hình để thêm dịch vụ PbsManagedJobFactoryService vào trong các dịch vụ của Globus Toolkit.

```
$GLOBUS_LOCATION/setup/globus/setup-pbs-provider
```

lệnh này thực hiện các cấu hình để thành phần RIPS (Resource Information Provider) trong lưới làm việc với PBS, RIPS nhận các thông tin trả về từ PBS_Server và cung cấp cho trình quản lý công việc.

```
$GLOBUS_LOCATION/setup/globus/setup-globus-job-manager-pbs --
cpu-per-node=1
```

lệnh này thiết lập số lượng các bộ vi xử lý (CPU) trên một nút đơn trong cluster, thông số này phải phù hợp với tài nguyên tương ứng trong cluster.

```
$GLOBUS_LOCATION/setup/globus/setup-globus-job-manager-pbs --
remote-shell=rsh
```

lệnh này thực hiện việc chuyển đổi remote-shell mặc định thành rsh, thông thường các hệ thống tính toán song song phân cụm thường dùng rsh để truy cập các tài nguyên trong cluster.

```
$GLOBUS_LOCATION/setup/globus/setup-globus-job-manager-pbs –
validate-queues=yes
```

lệnh này thực hiện kiểm tra hàng đợi tương ứng trong PBS phù với yêu cầu hay không.

4.4 Kết chương

Trong chương 4, luận văn đã giới thiệu các bước cần thực hiện trong việc lập kế hoạch triển khai một lưới. Đồng thời cũng đưa ra một mô hình lưới thử nghiệm, bao gồm cả các hướng dẫn cài đặt. Đây là bước áp dụng thực tế cho phần lý thuyết được nghiên cứu trong 3 chương đầu tiên.

KẾT LUẬN VÀ KIẾN NGHỊ

Kết quả thu được

Trong quá trình tìm hiểu về tính toán lưới và thực hiện cài đặt một lưới thử nghiệm, luận văn đã thu được các kết quả sau:

- Bước đầu làm chủ công nghệ tính toán lưới còn rất mới ở nước ta
- Nắm bắt được bộ công cụ Globus Toolkit và vận dụng vào cài đặt thực tế
- Triển khai hệ thống lưới thử nghiệm
- Biết cách kết nối cluster vào lưới, nhằm tăng sức mạnh tính toán cho lưới

Hướng phát triển

Có thể nói việc phát triển môi trường tính toán Mạng lưới là một hướng nghiên cứu công nghệ có triển vọng rất lớn trong tương lai, có những ứng dụng lớn lao để có thể tận dụng hết được nguồn lực tài nguyên rất đa dạng và phong phú hiện nay. Các vấn đề mà Grid đặt ra đã và đang được cộng đồng thế giới giải quyết. Các thành quả mà Gird Computing đem lại đã giúp giai quyết được nhiều bài toán mà trước đây không thể giải quyết bằng kỹ thuật toán song song.

Tuy nhiên cũng có một số bài toán mà công nghệ Grid chưa thể thể giải quyết được. Dù sao việc tận dụng được nguồn tài nguyên rãnh rỗi để tính tóan các bài tóan đòi hỏi lượng tính tóan và tài nguyên cao luôn là một thế mạnh để phát triển tốt môi trường tính toán mạng lưới này.

Dựa trên kiến thức đã thu được trong quá trình làm luận văn. Trong thời gian tới, việc nghiên cứu tiếp về tính toán lưới sẽ đi theo một trong các hướng phát triển sau:

- Nghiên cứu hạ tầng truyền thông trong môi trường lưới, tìm hiểu đưa ra các giải pháp để kết nối các nút lưới với nhau trong điều kiện hạ tầng truyền thông chưa tốt như ở Việt Nam.
- Về mặt lập trình thì tiến hành xây dựng các môdun riêng như: quản trị người dùng lưới, môi giới tài nguyên, lập lịch kinh tế, ….

TÀI LIỆU THAM KHẢO

Sách tiếng Anh:

[1] Ian Foster, Carl Kesselman (eds) - The Grid: Blueprint for a New Computing Infrastructure, 1st edition, Morgan Kaufmann Publishers, San Francisco, USA (1 November 1998), ISBN: 1558604758.

[2] Daniel Minoli - A Networking Approach to Grid Computing

[3] Ahmar Abbas - Grid Computing: A Practical Guide to Technology and Applications

[4] Fran Berman, Anthony J.G.Hey, Geoffrey C.Fox – Grid computing: Make the Global Infrastructure a Reality

[5] Globus Books - Globus Toolkit 4 Admin Guide

[6] Globus Books - A Globus Primer 0.6 (2005)

[7] IBM Red Books – Globus Toolkit 3.0 Quick Start

[8] Joshy Joseph, Craig Fellenstein - Grid Computing

[9] IBM Red Books – Fundamentals of Grid Computing.

[10] IBM Red Books – Introduction to Grid Computing with Globus Toolkit

[11] The Globus Alliance - How to Build a Service using GT4

[12] Sun BluePrints Online - The Sun Infrastructure Solution for Grid Computing

[13] Sun BluePrints Online - Introduction to the Cluster Grid

[14] Proytable Batch System - OpenPBS Release 2.3 - Administrator

[15] James Patton Jones - PBS Pro 5.0 - Administrator Guide

[16] Jarek Nabryski, Jenifer M.Schopf, Jan Weglars - Grid Resource Management

Sách tiếng Việt:

[17] Đồ án đại học - Kết nối Globus và PBS - Nguyễn Đức Mạnh khoá K45 trường ĐHBKHN

Trang Web:

[18] Workspace Admin Guide:http://workspace.globus.org/vm/TP1.2/doc/admin-index.html#installing-workspace-service

[19] SimpleCA: http://gdp.globus.org/gt3-tutorial/multiplehtml/ch12s02.html

[20] Website Globus Toolkits http://www.globus.org

[21] Globus Toolkit Tutorial: http://www.globusconsortium.org/tutorial/

[22] http://www.globus.org/toolkit/docs/4.0/admin/docbock/index.html

[23] http://www.globus.org/toolkit/docs/4.0/data/gridftp/admin-index.html

[24] http://www.globus.org/toolkit/docs/development/4.1.0/admin/docbook/quick start.html#q-container

[25] ML. Levisetto - About PBS and the Grid: http://www.bo.infn.it/alice/introgrid/pbsabout/index.html

[26] Quick Tutorial for PBS: http://www.princeton.edu/~letchu/compting/scientific_computing/PBS.html

[27] PBS Userguide: http://www.doesciencegrid.org/public/pbs/homepage.html

[28] http://www.w3.org/2002/ws/

[29] J. Postel, J. Reynolds, *"File Transfer Protocol",* tài nguyên sẵn có trên web tại địa chỉ http://www.w3.org/Protocols/rfc959/

[30] http://doesciencegrid.org/projects/GPDK/