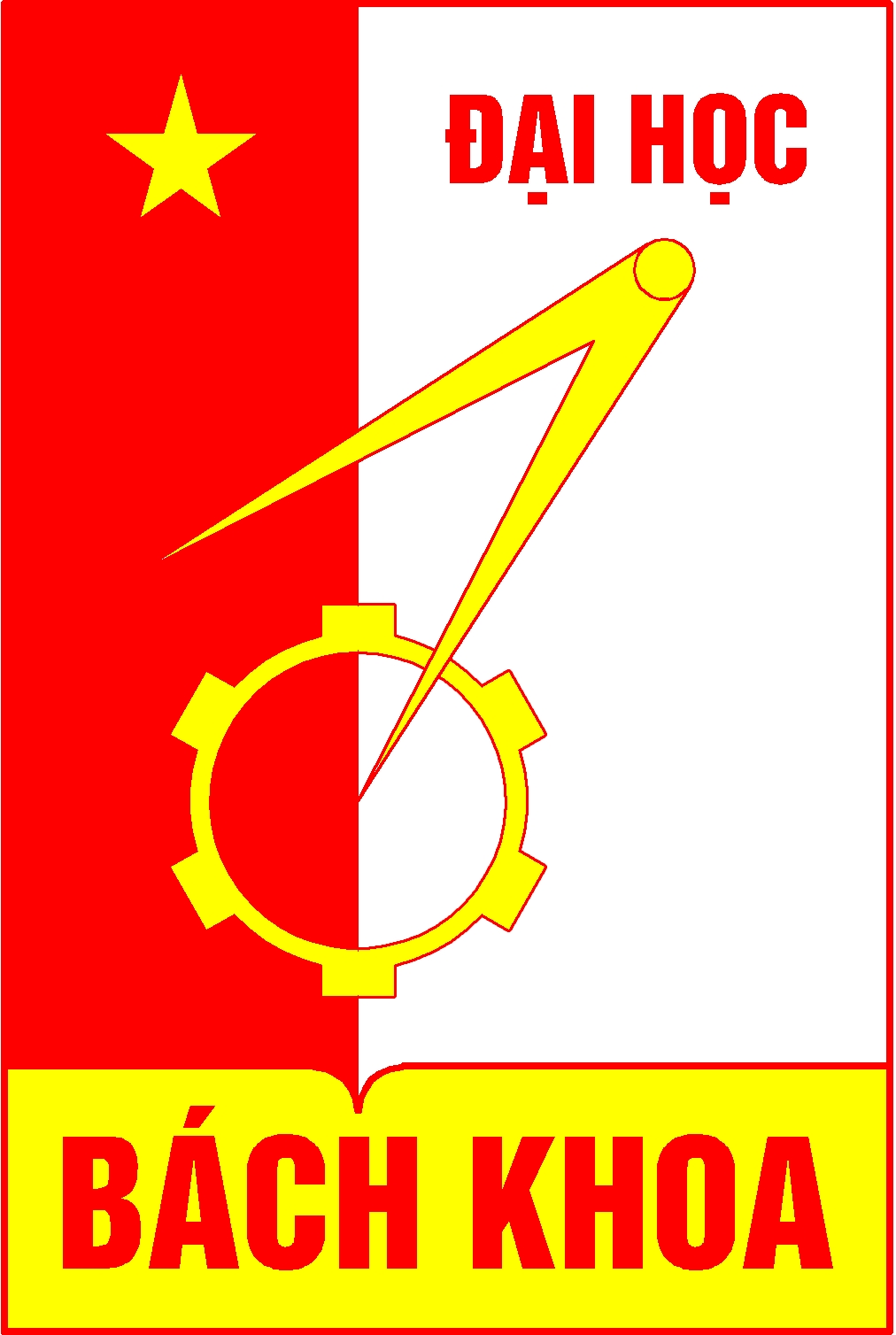


***Hà Nội, 12/2015***

**BÁO CÁO**

**ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**



**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

***Đề tài:* Khai thác phần mềm quản lý và giám sát mạng Cacti. Viết Plugin ghi nhật ký.**

***Giảng viên hướng dẫn*: Thầy Hồ Sỹ Bàng**

***Sinh viên thực hiện*: Trần Viết Phước**

**Lớp: Truyền thông mạng – K53**

**SHSV: 20082044**

**TÓM TẮT NỘI DUNG ĐỒ ÁN**

# TÓM TẮT NỘI DUNG ĐỒ ÁN

Thế giới hiện nay, khoa học công nghệ phát triển như vũ bão, đặc biệt là ngành công nghệ thông tin. Nó đã góp phần quan trọng trong cuộc sống hàng ngày của con người. Sự phát triển của Internet đã làm thay đổi các thói quen sinh hoạt, lao động sản xuất, mang lại lợi ích to lớn trong việc phát triển kinh tế xã hội, thông tin liên lạc của con người.

Internet phát triển với hàng trăm triệu máy tính được kết nối, sử dụng hàng ngày. Việc quản lý, giám sát các máy tính đó cũng vô cùng khó khăn. Vì vậy việc sử dụng các phần mềm quản lý mạng là rất cần thiết, trong đó có phần mềm Cacti giúp cho việc quản lý dễ dàng và đảm bảo hơn.

Để nhận thấy tầm quan trọng của các phần mềm quản lý mạng nói chung và Cacti nói riêng, em đã tìm hiểu, cài đặt và khai thác phần mềm mã nguồn mở Cacti để thực hiện việc quản lý các hệ thống mạng.

Cacti cũng có rất nhiều Plugin để mở rộng thêm các chức năng cho việc giám sát và quản lý hiệu quả hơn. Em đã thử viết Plugin ghi nhật ký gồm có các chức năng sau:

* Monitor: Quản lý các máy client, hiển thị các thông số dưới dạng bảng, đồ thị. Cảnh báo bằng email khi các thông số vượt ngưỡng cho phép.
* Syslog: Bẫy các sự kiện từ các client, đưa vào syslog.

Cấu trúc của đồ án:

* Phần A: Cơ sở lý thuyết
* Phần B: Xây dựng plugin ghi nhật ký
* Phần C: Cài đặt thực nghiệm và đánh giá

**NỘI DUNG**

[TÓM TẮT NỘI DUNG ĐỒ ÁN 2](#_Toc437157645)

[LỜI CẢM ƠN 5](#_Toc437157646)

[DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ 6](#_Toc437157647)

[DANH MỤC HÌNH VẼ 7](#_Toc437157648)

[ĐẶT VẤN ĐỀ 8](#_Toc437157649)

[PHẦN A: CƠ SỞ LÝ THUYẾT 9](#_Toc437157650)

[I.Tổng quan về quản trị mạng 9](#_Toc437157651)

[1.Khái niệm quản trị mạng 9](#_Toc437157652)

[2. SNMP–Simple Network ManageProtocol 10](#_Toc437157653)

[2.1.Giới Thiệu về Giao thức SNMP 10](#_Toc437157654)

[2.2.Ưu điểm sử dụng giao thức SNMP 10](#_Toc437157655)

[2.3.Các phiên bản giao thức SNMP 10](#_Toc437157656)

[2.4.Các thành phần trong SNMP 13](#_Toc437157657)

[2.5.Các cơ chế bảo mật SNMP 19](#_Toc437157658)

[II.Hệ thống giám sát và quản lý mạngCacti 20](#_Toc437157659)

[1.Các hệ thống giám sát và quản lý mạng 20](#_Toc437157660)

[1.1.Tổng quan 20](#_Toc437157661)

[1.2.Cấu trúc của một hệ quản lý mạng 20](#_Toc437157662)

[1.3.Các chức năng hệ thống NMS 21](#_Toc437157663)

[2.Cacti 22](#_Toc437157664)

[2.1.Giới thiệu 22](#_Toc437157665)

[2.2.Kiến trúc và hoạt động của Cacti. 22](#_Toc437157666)

[2.3. Các chức năng chính của Cacti 25](#_Toc437157667)

[2.4.Kiến trúc plugin Cacti 27](#_Toc437157668)

[Phần B: PLUGIN GHI NHẬT KÝ 36](#_Toc437157669)

[I.Yêu cầu, ý tưởng và giải pháp 36](#_Toc437157670)

[1.Yêu cầu 36](#_Toc437157671)

[2.Plugin ghi nhật ký 36](#_Toc437157672)

[II.Thiết kế hệ thống 36](#_Toc437157673)

[1.Mô tả chương trình 36](#_Toc437157674)

[2.Mô tả hệ thống 36](#_Toc437157675)

[3.Thiết kế cơ sơ dữ liệu 36](#_Toc437157676)

[4.Các module và chức năng chính 36](#_Toc437157677)

[Phần C: CÀI ĐẶT THỰC NGHIỆM VÀ ĐÁNH GIÁ 37](#_Toc437157678)

[I.Cài đặt và cấu hình 37](#_Toc437157679)

[1.Cài đặt 37](#_Toc437157680)

[1.1.Cài đặt Cacti 37](#_Toc437157681)

[1.2.Cài đặt Plugin 44](#_Toc437157682)

[2.Cấu hình 44](#_Toc437157683)

[2.1.Cấu hình trên Clients. 44](#_Toc437157684)

[2.2.Cấu hình trên localhost 47](#_Toc437157685)

[II.Thực nghiệm 51](#_Toc437157686)

[III.Kết quả thực nghiệm 51](#_Toc437157687)

[IV.Đánh giá 51](#_Toc437157688)

[1.Ưu điểm 51](#_Toc437157689)

[2.Nhược điểm 51](#_Toc437157690)

[3.Hướng phát triển 51](#_Toc437157691)

[KẾT LUẬN 52](#_Toc437157692)

[TÀI LIỆU THAM KHẢO 53](#_Toc437157693)

# LỜI CẢM ƠN

Trước hết, em xin gửi lời cảm ơn chân thành tới Thầy Hồ Sỹ Bàng, người đã tận tình dạy dỗ và hướng dẫn em trong quá trình hoàn thành đồ án cũng như trong học tập.

Đồng thời, em xin bày tỏ lòng biết ơn đến các thầy cô giáo trong Viện Công nghệ thông tin và Truyền thông – trường Đại học Bách Khoa Hà Nội, những người đã tận tâm giảng dạy, truyền đạt cho chúng em những kiến thức cơ bản làm nền tảng cho việc thực hiện đồ án cũng như trong quá trình công tác sau này.

Em cũng xin gửi lời cảm ơn tới các anh chị tại trường Đại học Bách Khoa Hà Nội, các bạn, các em trong nhóm sinh viên nghiên cứu, những người luôn ở bên cạnh giúp đỡ, động viên em trong quá trình hoàn thành đồ án.

Cuối cùng, với tất cả sự kính trọng, con xin bày tỏ lòng biết ơn sâu sắc tới bố mẹ và anh chị em trong gia đình đã luôn là chỗ dựa tinh thần vững chắc và tạo mọi điều kiện cho con ăn học nên người.

Hà Nội, ngày 28 tháng 11 năm 2015

Trần Viết Phước

# DANH MỤC CÁC TỪ VIẾT TẮT VÀ THUẬT NGỮ

|  |  |  |
| --- | --- | --- |
| Chữ viết tắt | Viết đầy đủ | Ý nghĩa |
| **SNMP** | Simple Network Management Protocol | Giao thức quản lý mạng đơn giản |
| **RFC** | Request for Comments | Chuỗi các bản ghi nhớ chứa nghiên cứu về Internet |
| **TCP/IP** | Transmission Control Protocol/Internet Protocol | Bộ giao thức truyền thông |
| **NMS** | Network Management Station (Network Monitoring System ) | Hệ thống giám sát, quản lý mạng |
| **OID** | Object ID | Đối tượng lưu các thông tin thiết bị mạng trên nền SNMP |
| **MIB** | Management Information Base | Cấu trúc dữ liệu các thông tin của các thiết bị chạy trên nền TCP/IP |
| **ACL** | Access Control List | Danh sách các địa chỉ IP được quản lý |
| **RRDTool** | Round-Robin Database Tool | Hệ thống lưu trữ và hiển thị thời gian, dữ liệu, băng thông mạng,.. |

# DANH MỤC HÌNH VẼ

*Hình 1. Các phương thức trong SNMPv1*

*Hình 2. Mối quan hệ giữa ba thành phần SNMP*

*Hình 3. Mô hình giao thức hoạt dộng SNMP*

*Hình 4. Vị trí của SNMP trong chồng giao thức TCP/IP*

*Hình 5. Sơ đồ khối kiến trúc Cacti*

*Hình 6. Hình ảnh dạng đồ thị kết quả của Cacti*

*Hình 7. So sánh Cacti với các hệ thống NMS khác*

*Hình 8. Cơ chế hoạt động của plugin*

*Hình 9. Thành phần của Plugin Architecture*

*Hình 10. Tương tác giữa Plugin Architecture và các thành phần trong Cacti*

*Hình 11. Quan hệ giữa Plugin Architecture và Plugin*

*Hình 12. Các Hook API*

# ĐẶT VẤN ĐỀ

Một trong những công việc cơ bản của quả trị mạng là giám sát mạng để theo dõi hoạt động của toàn bộ hệ thống mạng. Giám sát mạng sẽ giúp thu thập thông tin định kỳ về hệ thống mạng, giúp thu thập và hiển thị các thông tin thống kế, biểu đồ hiệu suất của hệ thống. Đây là cơ sở phục vụ cho tối ưu hóa cơ sở hạ tầng mạng và hiệu suất của hệ thống.

Phần mềm giám sát và quản lý mạng Cacti là một trong những công cụ hữu hiệu phục vụ giám sát hệ thống mạng. Với vài trò là phần mềm mã nguồn mở Cacti cho phép người quản trị có thể thêm nhiều plugin, tác động trực tiếp lên phần mềm để công việc theo dõi phù hợp với hệ thống và thuận lợi nhất.

Chính vì vậy, việc xây dựng plugin ghi nhật ký để người quản trị có thể theo dõi, giám sát được tất cả các thông số, các sự kiện của các máy tính trong hệ thống và của cả hệ thống sẽ giúp người quản trị có thể kiểm soát, quản lý toàn bộ hệ thống một cách chính xác và dễ dàng.

# PHẦN A: CƠ SỞ LÝ THUYẾT

# I.Tổng quan về quản trị mạng

## 1.Khái niệm quản trị mạng

Một hệ quản trị mạng cần phải có 5 chức năng sau:

* *Quản lý hiệu năng (Performance Management)*: Đo lường, quản lý các khía cạnh khác nhau của hiệu năng mạng: băng thông mạng, thời gian hồi đáp người dùng ... Mục tiêu của nó là duy trì hiệu năng của mạng ở mức có thể truy cập được.

Quy trình quản lý hiệu năng bao gồm các giai đoạn: 1, Thu thập dữ liệu hiệu năng. 2, Phân tích dữ liệu. 3, Thiết lập các ngưỡng hiệu năng cho giá trị của các thông số quan trọng.

* *Quản lý cấu hình (Configuration Management)*: Theo dõi và thu thập thông tin cấu hình hệ thống của mạng, của phần cứng và cả phần mềm. Thông tin thu thập được thường được lưu trữ vào CSDL để sao cho dễ dàng truy cập và phân tích.
* *Quản lý lỗi (Fault Management)*: Phát hiện lỗi, ghi nhận lỗi, thông báo lỗi và đặc biệt có thể tự động khắc phục một số vấn đề về mạng để bảo đảm cho mạng có thể duy trì hiệu quả ở mức có thể chấp nhận được.

Quy trình xử lý lỗi bao gồm các giai đoạn: 1, Xác định “triệu chứng” và cô lập vấn đề căn nguyên của lỗi. 2, Khắc phục lỗi và kiểm tra sự hoạt động trở lại bình thường của hệ thống. 3, Ghi lại vấn đề/ lỗi được phát hiện và giải pháp khắc phục chúng (chuẩn bị cho việc khắc phục lỗi lần sau).

* *Quản lý tài khoản (Accounting Management)*: Chuẩn bị số liệu để báo cáo về việc sử dụng mạng và cấp phát tài nguyên của toàn hệ thống. Quy trình: 1, Đo lường việc sử dụng của tất cả các tài nguyên mạng quan trọng. 2, Phân tích các mẫu sử dụng hiện tại. 3, Thiết lập hạn ngạch sử dụng tài nguyên mạng cho các nhóm user khác nhau.

Mục đích của quản lý kiểm toán là làm sao cho tài nguyên hệ thống được khai thác và sử dụng một cách hiệu quả nhất.

* *Quản lý bảo mật (Security Management)*: Điều khiển truy cập đến các tài nguyên mạng theo đặc quyền của người dùng, nhằm bảo vệ các tài nguyên nhạy cảm, ngăn chặn sự truy cập bất hợp lệ của người sử dụng không được phép.

Quy trình bảo mật liên quan đến: 1, Định danh các tài nguyên nhạy cảm. 2, Xác định mối liên quan giữa tài nguyên nhạy cảm và nhóm user được quyền/không được quyền truy cập vào nhóm tài nguyên đó. 3, Theo dõi việc truy cập đến tài nguyên nhạy cảm của user.

Có các phương pháp kỹ thuật quản lý mạng:

1. *Quản lý tập trung*
2. *Quản lý phân cấp*
3. *Quản lý phân tán*
4. *Quản lý lai ghép*
5. *Quản lý hướng đối tượng*
6. *Mô hình quản lý mạng theo tiêu chuẩn ISO*

## 2. SNMP–Simple Network ManageProtocol

### 2.1.Giới Thiệu về Giao thức SNMP

Vào đầu năm 1988, Tổ chức kiến trúc Internet IAB (Internet Architecture Board) nhận thấy sự cần thiết có bộ công cụ quản lí cho TCP/IP nên đã cho ra đời RFC 1052.RFC 1052 là các yêu cầu tiêu chuẩn hoá quản lí mạng và tậptrung vào các vấn đề quản lí mạng phải thực hiện:

* Đảm bảo tính mở rộng
* Đảm bảo tính đa dạng để phát triển
* Đảm bảo tính đa dạng trong quản lí
* Bao trùm nhiều lớp giao thức

Tháng 4 năm 1993, SNMPv2 trở thành tiêu chuẩn quản lí mạng đơn giản thay thế SNMPv1. SNMPv2 bổ sung một số vấn đề mà SNMPv1 còn thiếu như nhận thực và bảo mật. Tuy nhiên, SNMPv2 khá phức tạp và khó tương thích với SNMPv1.

Năm 1997, SNMPv3 ra đời nhằm tương thích với các giao thức đa phương tiện trong quản lí mạng, phát triển trên nền java và đưa ra kiến trúc và giao thức mới như giao thức quản lí đa phương tiện HMMP (Hypermedia Management Protocol).

### 2.2.Ưu điểm sử dụng giao thức SNMP

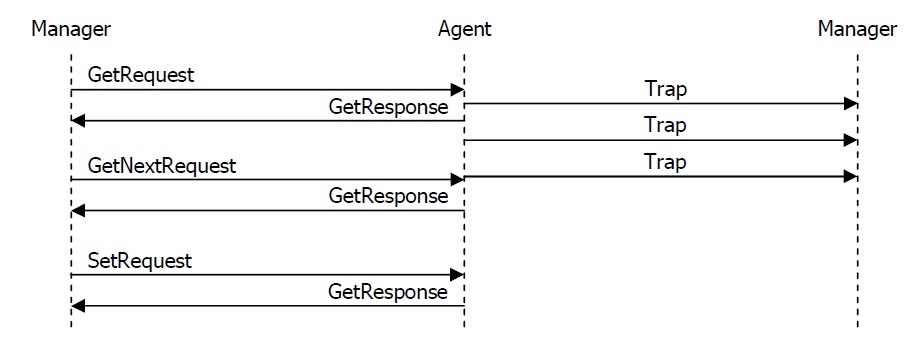
SNMP được thiết kế để đơn giản hóa quá trình quản lý các thành phần trong mạng. Nhờ đó các phần mềm SNMP có thể được phát triển nhanh và tốn ít chi phí. SNMP được thiết kế để có thể mở rộng các chức năng quản lý, giám sát. Khi có một thiết bị mới với các thuộc tính, tính năng mới thì người ta có thể thiết kế tùy chọn SNMP để phục vụ cho riêng mình. SNMP được thiết kế để có thể hoạt động độc lập với các kiến trúc và cơ chế của các thiết bị hỗ trợ SNMP.Các thiết bị khác nhau có hoạt động khác nhau nhưng hoạt động dựa trên giao thức SNMP là giống nhau.

### 2.3.Các phiên bản giao thức SNMP

Hiện tạiSNMP có 3 phiên bản : SNMPv1, SNMPv2 và SNMPv3. Các phiên bản này khác nhau một chút ở định dạng bản tin và phương thức hoạt động. Hiện nay SNMPv1 là phổ biến nhất do có nhiều thiết bị tương thích nhất và có nhiều phần mềm hỗ trợ nhất. Trong khi đó chỉ có một số thiết bị và phần mềm hỗ trợ SNMPv3.

*-****Phiên bản SNMPv1***: phiên bản đầu tiên của SNMP, có 5 phương thức Get, GetNext, Set, Response, Trap.

* *GetRequest* : Bản tin GetRequest được manager gửi đến agent để lấy một thông tin nào đó. Trong GetRequest có chứa ID của object muốn lấy. Ví dụ: Muốn lấy thông tin tên của Device1 thì manager gửi bản tin GetRequest ID=1.3.6.1.2.1.1.5 đến Device1, tiến trình SNMP agent trên Device1 sẽ nhận được bản tin và tạo bản tin trả lời. Trong một bản tin GetRequest có thể chứa nhiều OID, nghĩa là dùng một GetRequest có thể lấy về cùng lúc nhiều thông tin.
* *GetNextRequest:* Bản tin GetNextRequest cũng dùng để lấy thông tin và cũng có chứa OID, tuy nhiên nó dùng để lấythông tin của object nằm kế tiếp object được chỉ ra trong bản tin. Chúng ta đã biết khi đọc qua những phần trên: một MIBbao gồm nhiều OID được sắp xếp thứ tự nhưng không liên tục, nếu biết một OID thì không xác định đượcOID kế tiếp. Do đó ta cần GetNextRequest để lấy về giá trị của OID kế tiếp. Nếu thực hiện GetNextRequestliên tục thì ta sẽ lấy được toàn bộ thông tin của agent.
* *SetRequest:* Bản tin SetRequest được manager gửi cho agent để thiết lập giá trị cho một object nào đó. Ví dụ: Có thể đặt lại tên của một máy tính hay router bằng phần mềm SNMP manager, bằng cách gửi bảntin SetRequest có OID là 1.3.6.1.2.1.1.5.0 (sysName.0) và có giá trị là tên mới cần đặt.
* *GetResponse:* Mỗi khi SNMP agent nhận được các bản tin GetRequest, GetNextRequest hay SetRequest thì nó sẽ gửi lại bản tin GetResponse để trả lời. Trong bản tin GetResponse có chứa OID của object được request và giá trị của object đó.
* *Trap*: Bản tin Trap được agent tự động gửi cho manager mỗi khi có sự kiện xảy ra bên trong agent, các sự kiệnnày không phải là các hoạt động thường xuyên của agent mà là các sự kiện mang tính biến cố. Ví dụ: Khicó một port down, khi có một người dùng login không thành công, hoặc khi thiết bị khởi động lại, agent sẽgửi trap cho manager.Tuy nhiên không phải mọi biến cố đều được agent gửi trap, cũng không phải mọi agent đều gửi trap khixảy ra cùng một biến cố. Việc agent gửi hay không gửi trap cho biến cố nào là do hãng sản xuấtdevice/agent quy định.



*Hình 1. Các phương thức trong SNMPv1*

***- Phiên bản SNMPv2***: SNMPv2 tích hợp khả năng liên điều hành từ manager tới manager và hai đơn vị dữ liệu giao thức mới. Khả năng liên kết điều hành manager-manager cho phép SNMP hỗ trợ quản lí mạng phân tán trong một trạm và gửi báo cáo tới một trạm khác. Hai đơn vị dữ liệu giao thức PDU (Protocol Data Unit) là GetbulkRequest và InformRequest. Các PDU này liên quan tới xử lý lỗi và khả năng đếm của SNMPv2. Khả năng đếm trong SNMPv2 sử dụng bộ đếm 64 bit (hoặc 32) để duy trì trạng thái của các liên kết và giao diện.

*MIB cho SNMPv2*: MIB trong SNMPv2 định nghĩa các đối tượng mô tả tác động của một phần tử SNMPv2.MIB gồm 3 nhóm:

Nhóm hệ thống (System group): là một mở rộng của nhóm system trong MIB-II gốc, bao gồm một nhóm các đối tượng cho phép một Agent SNMPv2 mô tả các đối tượng tài nguyên của nó.

Nhóm SNMP (SNMP group): một cải tiến của nhóm SNMP trong MIB-II gốc, bao gồm các đối tượng cung cấp các công cụ cơ bản cho hoạt động giao thức.

Nhóm các đối tượng MIB (MIB objects group): một tập hợp các đối tượng liên quan đến các SNMPv2-trap PDU và cho phép một vài phần tử SNMPv2 cùng hoạt động, thực hiện như trạm quản trị, phối hợp việc sử dụng của chúng trong toán tử Set của SNMPv2

Nhóm hệ thống: nhóm system định nghĩa trong SNMPv2 giống trong MIB-II và bổ sung một vài đối tượng mới.

Nhóm SNMP: Nhóm này gần giống như nhóm SNMP đươc định nghĩa trong MIB-II nhưng có thêm một số đối tượng mới và loại bỏ một số đối tượng ban đầu.

Nhóm SNMP chứa một vài thông tin lưu lượng cơ bản liên quan đến toán tử SNMPv2 và chỉ có một trong các đối tượng là bộ đệm chỉ đọc 32-bit.

Nhóm đối tượng MIB: Nhóm các đối tượng MIB chứa các đối tượng thích hợp thêm vào việc điều khiển các đối tượng MIB.

**- *Phiên bản SNMPv3***: SNMPv3 dựa trên việc thực hiện giao thức, loại dữ liệu và uỷ quyền như SNMPv2 và cải tiến phần an toàn. SNMPv3 cung cấp an toàn truy nhập vào các thiết bị bằng cách kết hợp sự xác nhận và mã khoá các gói tin trên mạng. Những đặc điểm bảo mật cung cấp trong SNMPv3 là:

Tính toàn vẹn thông tin : Đảm bảo các gói tin không bị sửa trong khi truyền.

Sự xác nhận: Xác nhận nguồn của thông tin gửi đến.

Mã khoá: Đảo nội dung của gói tin, ngăn cản việc gửi thông báo từ nguồn không được xác nhận.

Tuy nhiên việc sử dụng SNMPv3 rất phức tạp và cồng kềnh dù nó là sự lựa chọn tốt nhất cho vấn đề bảo mật của mạng. Việc sử dụng sẽ tốn rất nhiều tài nguyên do trong mỗi bản tin truyền đi sẽ có phần mã hóa BER. Phần mã hóa này sẽ chiếm một phần băng thông đường truyền do đó làm tăng phí tổn mạng. Mặc dù được coi là phiên bản đề nghị cuối cùng và được coi là đầy đủ nhất nhưng SNMPv3 vẫn chỉ là tiêu chuẩn dự thảo và vẫn đang được nghiên cứu hoàn thiện.

Khuôn dạng bảng tin SNMPv3: RFC 2572 định nghĩa các khuôn dạng bản tin SNMPv3. Khuôn dạng bản tin SNMPv3 được phân chia trong trong bốn phần

Dữ liệu chung (Common data)- Trường này xuất hiện trong tất cả các bản tin SNMPv3.

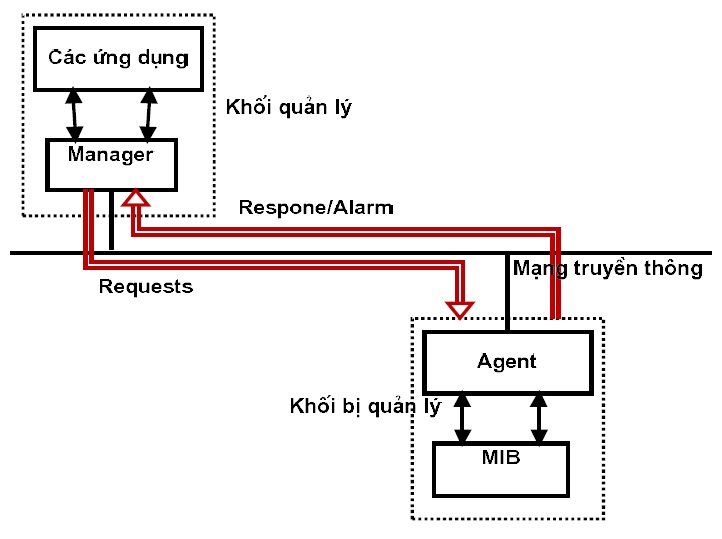
Bảo mật mô hình dữ liệu (Security model data)- Vùng này có ba phần: phần chung, phần dành cho sự chứng thực và phần cho dữ liệu riêng.

Context – Hai trường nhận dạng và tên được dùng để cung cấp context cho PDU nào sẽ phải xử lý.

PDU –Vùng này chứa một SNMPv2c PDU.

### 2.4.Các thành phần trong SNMP

Hệ thống quản lý mạng dựa trên SNMP gồm ba thành phần: bộ phận quản lý (manager), thiết bị chịu sự quản lý – còn gọi là đại lý (agent) và cơ sở dữ liệu gọi là Cơ sở thông tin quản lý (MIB). Mặc dù SNMP là một giao thức quản lý việc chuyển giao thông tin giữa ba thực thể trên, song nó cũng định nghĩa mối quan hệ client-server (chủ tớ). Cơ sở dữ liệu do agent SNMP quản lý là đại diện cho MIB của SNMP. Minh họa mối quan hệ giữa ba thành phần SNMP này.

*Hình 2. Mối quan hệ giữa ba thành phần SNMP*

*Hình 1.9:**Mối quan hệ giữa các thành phần SNMP*

**a.Bộ phận quản lý (manager)**

Bộ phận quản lý là một chương trình vận hành trên một hoặc nhiều máy tính trạm.

Tùy thuộc vào cấu hình, mỗi bộ phận quản lý có thể được dùng để quản lý một mạng con, hoặc nhiều bộ phận quản lý có thể được dùng để quản lý cùng một mạng con hay một mạng chung. Tương tác thực sự giữa một người sử dụng cuối (end-user) và bộ phận quản lý được duy trì qua việc sử dụng một hoặc nhiều chương trình ứng dụng mà, cùng với bộ phận quản lý, biến mặt bằng phần cứng thành trạm quản lý mạng (NMS). Ngày nay, trong thời kỳ các chương trình giao diện người sử dụng đồ họa (GUI), hầu hết những chương trình ứng dụng sẽ cho ra giao diện sử dụng con trỏ và chuột để phối hợp hoạt động với bộ phận quản lý tạo ra những bản đồ họa và biểu đồ cung cấp những tổng kết hoạt động của mạng dưới dạng thấy được. Qua bộ phận quản lý, những yêu cầu được chuyển tới một hoặc nhiều thiết bị chịu sự quản lý ban đầu SNMP được phát triển để sử dụng trên mạng TCP/IP và những mạng này tiếp tục làm mạng vận chuyển cho phần lớn các sản phẩm quản lý mạng dựa trên SNMP. Tuy nhiên SNMP cũng có thể được chuyển qua NetWare IPX và những cơ cấu vận chuyển khác.

**b.Agent**

Thiết bị chịu sự quản lý (Agent) là một nút mạng hỗ trợ giao thức SNMP và thuộc về mạng bị quản lý. Thiết bị có nhiệm vụ thu thập thông tin quản lý và lưu trữ để phục vụ cho hệ thống quản lý mạng. Những thiết bị chịu sự quản lý, đôi khi được gọi là những phần tử mạng, có thể là các bộ định tuyến và máy chủ truy nhập (Access Server), switch, bridge, hub và máy tính hay là máy in trong mạng. Mỗi thiết bị chịu sự quản lý bao gồm phần mềm hoặc phần sụn (firmware) dưới dạng mã phiên dịch những yêu cầu SNMP và đáp ứng của những yêu cầu đó. Phần mềm hoặc phần sụn này được coi là một agent. Mặc dù mỗi thiết bị bắt buộc bao gồm một agent chịu quản lý trực tiếp, những thiết bị không tương thích với SNMP cũng có thể quản lý được nếu như chúng hỗ trợ một giao thức quản lý độc quyền. Để thực hiện được điều này phải có agent ủy nhiệm (proxy agent). Proxy agent này có thể được coi như một bộ chuyển đổi giao thức vì nó phiên dịch những yêu cầu SNMP thành giao thức quản lý độc quyền của thiết bị không hoạt động theo giao thức SNMP. Mặc dù SNMP chủ yếu là giao thức đáp ứng thăm dò (poll-respond) với những yêu cầu do bộ phận quản lý tạo ra dẫn đến những đáp ứng trong agent, agent cũng có khả năng đề xướng ra một “đáp ứng tự nguyện”. Đáp ứng tự nguyện này là điều kiện cảnh báo từ việc giám sát agent với hoạt động đã được định nghĩa trước và đáp ứng này cảnh báo việc agent đã tới ngưỡng định trước. Dưới sự điều khiển SNMP, việc truyền cảnh báo này được gọi là cái bẫy (TRAP).

**c.Cơ sở thông tin quản lý – MIB**

Mỗi thiết bị chịu sự quản lý có thể có cấu hình, trạng thái và thông tin thống kê định nghĩa chức năng và khả năng vận hành của thiết bị. Thông tin này rất đa dạng, có thể bao gồm việc thiết lập chuyển mạch phần cứng, những giá trị khác nhau lưu trữ trong các bảng ghi nhớ dữ liệu, bộ hồ sơ hoặc các trường thông tin trong hồ sơ lưu trữ ở các file và những biến hoặc thành phần dữ liệu tương tự. Nhìn chung, những thành phần dữ liệu này được coi là Cơ sở thông tin quản lý của thiết bị chịu sự quản lý. Xét riêng, mỗi thành phần dữ liệu biến đổi được coi là một đối tượng bị quản lý và bao gồm tên, một hoặc nhiều thuộc tính và một tập các hoạt động (operation) thực hiện trên đối tượng đó. Vì vậy MIB định nghĩa loại thông tin có thể khôi phục từ một thiết bị chịu sự quản lý và cách cài đặt thiết bị mà hệ thống quản lý điều khiển.

**d.Mô hình giao thức SNMP**

SNMP sử dụng các dịch vụ chuyển tải dữ liệu thông qua các giao thức UDP/IP. Một ứng dụng của Manager phải nhận dạng được Agent cần thông tin với nó. Một ứng dụng của Agent được nhận dạng bởi địa chỉ IP của nó và một cổng UDP. Một ứng dụng Manager đóng gói yêu cầu SNMP trong một UDP/IP, UDP/IP chứa mã nhận dạng cổng nguồn, địa chỉ IP đích và mã nhận dạng cổng UDP của nó. Khung UDP sẽ được gửi đi thông qua thực thể IP tới hệ thống chịu sự quản lý, tại đó khung UDP sẽ được phân phối bởi thực thể UDP tới Agent. Tương tự, các bản tin TRAP phải được các Manager nhận dạng. Các bản tin sử dụng địa chỉ IP và mã nhận dạng cổng UDP của Manager SNMP. SNMP sử dụng 3 lệnh cơ bản là Read, Write, Trap và một số lệnh tùy biến để quản lý thiết bị

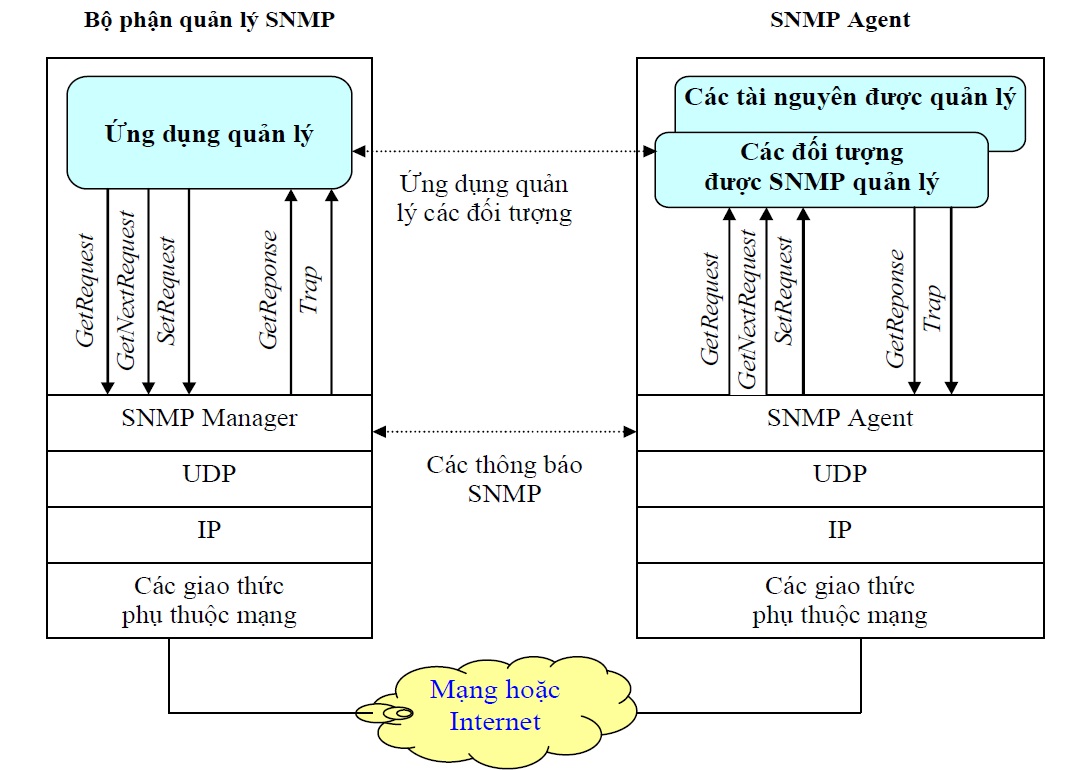
*Lệnh Read:* Được SNMP dùng để đọc thông tin từ thiết bị. Các thông tin này được cung cấp qua các biến SNMP lưu trữ trên thiết bị và được thiết bị cập nhật.

*Lệnh Write:* Được SNMP dùng để ghi các thông tin điều khiển lên thiết bị bằng cách thay đổi giá trị các biến SNMP.

*Lệnh Trap:* Dùng để nhận các sự kiện gửi từ thiết bị đến SNMP. Mỗi khi có một sự kiện xảy ra trên thiết bị một lệnh Trap sẽ được gửi tới NMS.

SNMP điều khiển, theo dõi thiết bị bằng cách thay đổi hoặc thu thập thông tin qua các biến giá trị lưu trên thiết bị. Các Agent cài đặt trên thiết bị tương tác với những chip điều khiển hỗ trợ SNMP để lấy nội dung hoặc viết lại nội dung.

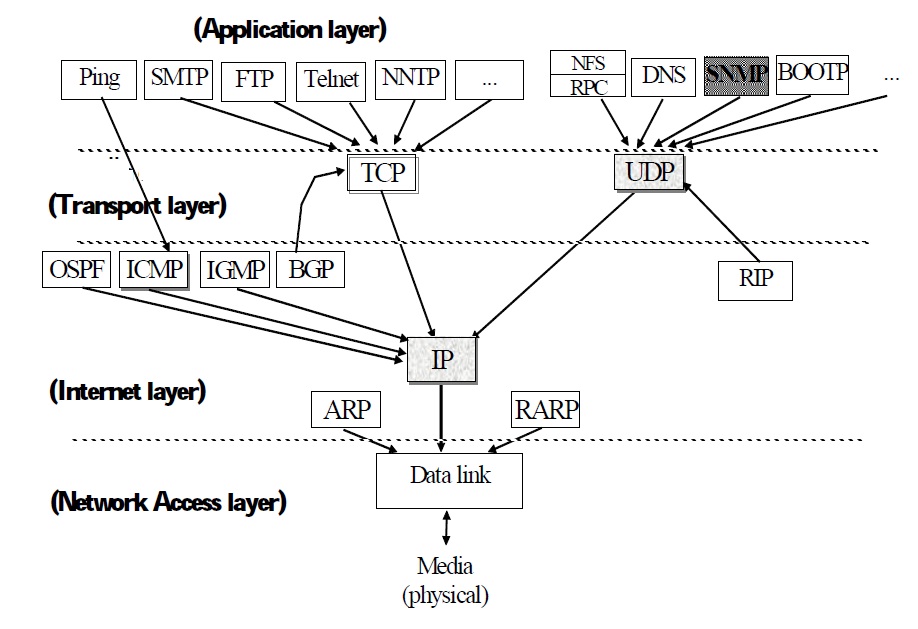
Giao thức SNMP sử dụng kiểu kết nối vô hướng (connectionless) để trao đổi thông tin giữa các phần tử và hệ thống quản lý mạng (cụ thể là UDP - User Datagram Protolcol - Giao thức dữ liệu đồ người sử dụng). UDP truyền các gói tin theo các khối riêng biệt. Tuy vậy có thể tùy ý sử dụng các giao thức khác để truyền các gói tin SNMP. Khi gửi các gói tin qua mạng, các phần tử mạng hay hệ thống quản lý mạng vẫn giữ nguyên định dạng của SNMP.



*Hình 3. Mô hình giao thức hoạt dộng SNMP*

Ta thấy, SNMP thuộc về lớp ứng dụng trong mô hình giao thức, nó sử dụng UDP làm giao thức lớp vận chuyển trên mạng IP.

*Quản lý liên lạc giữa manager với các agent:* Nhìn trên phương diện truyền thông, manager và các agent cũng là những người sử dụng, sử dụng một giao thức ứng dụng. Giao thức quản lý yêu cầu cơ chế vận chuyển để hỗ trợ tương tác giữa các agent và manager. Manager trước hết phải xác định được các agent mà nó muốn liên lạc. Có thể xác định được ứng dụng agent bằng địa chỉ IP của nó và cổng UDP được gán cho nó. Cổng UDP 161 được dành riêng cho các agent SNMP. Manager gói lệnh SNMP vào một tiêu đề UDP/IP. Tiêu đề này chứa cổng nguồn, địa chỉ IP đích và cổng 161. Một thực thể IP tại chỗ sẽ chuyển giao gói UDP tới hệ thống bị quản lý. Tiếp đó, một thực thể UDP tại chỗ sẽ chuyển phát nó tới các agent. Tương tự như vậy, lệnh TRAP cũng cần xác định những manager mà nó cần liên hệ. Chúng sử dụng địa chỉ IP cũng như cổng UDP dành cho SNMP manager, đó là cổng 162.



*Hình 4. Vị trí của SNMP trong chồng giao thức TCP/IP*

*Cơ chế vận chuyển thông tin giữa manager và agent:* Việc lựa chọn cơ chế vận chuyển là độc lập với giao thức truyền thông đó. SNMP chỉ đòi hỏi cơ chế vận chuyển không tin cậy dữ liệu đồ (datagram) để truyền đưa các PDU (đơn vị dữ liệu giao thức) giữa manager và các agent. Điều này cho phép sự ánh xạ của SNMP tới nhiều nhóm giao thức. Mô hình vận chuyển datagram giảm được độ phức tạp của ánh xạ tầng vận chuyển. Tuy nhiên, vẫn có một số lựa chọn cho tầng vận chuyển. Các tầng vận chuyển khác nhau có thể sử dụng nhiều kĩ thuật đánh địa chỉ khác nhau. Các tầng vận chuyển khác nhau có thể đưa ra những hạn chế quy mô của PDU. Ánh xạ tầng vận chuyển có trách nhiệm phải xử lý các vấn đề đánh địa chỉ, hạn chế quy mô PDU và một số tham số tầng vận chuyển khác. Trong phiên bản thứ hai của SNMP, người ta đã đơn giản hóa quá trình ánh xạ tới các chuẩn vận chuyển khác nhau. Giao thức quản lý được tách khỏi môi trường vận chuyển và điều này cũng được khuyến khích sử dụng cho bất cứ nhóm giao thức nào.

*Bảo vệ truyền thông liên lạc giữa manager và các agent khỏi sự cố:* Trong điều kiện mạng thiếu ổn định và tin cậy thì việc truyền thông quản lý càng trở nên quan trọng. Làm thế nào để các manager liên lạc với các agent một cách tin cậy? Việc SNMP sử dụng cơ chế UDP để liên lạc đã làm thiếu đi độ tin cậy vì UDP hoạt động theo kiểu dữ liệu đồ. SNMP để lại cho chương trình manager hoàn toàn chịu trách nhiệm và xử lý việc mất thông tin. Các lệnh GET, GET-NEXT và SET đều được phúc đáp bằng một lệnh GET-RESPONSE. Hệ thống có thể dễ dàng phát hiện ra việc bị mất một lệnh khi không nhận được lệnh trả lời. Nó có thể lặp lại yêu cầu đó một lần nữa hoặc có những hành động khác. Tuy nhiên, các bản tin TRAP do agent tạo ra lại không yêu cầu phúc đáp. Khi bị thất lạc bản tin TRAP, các chương trình agent sẽ không biết được điều đó (tất nhiên là manager cũng không hay biết về điều này). Thông thường các bản tin TRAP mang những thông tin hết sức quan trọng cho manager, do vậy manager cần chú ý và cần bảo đảm việc vận chuyển chúng một cách tin cậy. Một câu hỏi đặt ra là làm thế nào để vận chuyển mà tránh được mất mát, thất lạc các bản tin TRAP? Ta có thể thiết kế cho các agent gửi lặp lại bản tin TRAP. Biến số MIB có thể đọc số lần lặp lại theo yêu cầu. Lệnh SET của manager có thể đặt cấu hình cho biến số này. Có một cách khác là agent có thể lặp lại lệnh TRAP cho đến khi manager đặt biến số MIB để chấm dứt sự cố. Tuy nhiên, cả hai phương pháp trên đều chỉ cho ta những giải pháp từng phần. Trong trường hợp thứ nhất, số lần lặp lại có thể không đủ để đảm bảo liên lạc một cách tin cậy. Trong trường hợp thứ hai, một sự cố mạng có thể dẫn đến việc hàng loạt bản tin TRAP bị mất tùy thuộc vào tốc độ mà các agent tạo ra chúng. Điều này làm cho sự cố mạng trở nên trầm trọng hơn. Trong cả hai trường hợp, nếu ta cần chuyển những bản tin TRAP tới nhiều manager thì có thể xảy ra tình trạng không nhất quán giữa các manager hoặc xảy ra hiện tượng thất lạc thông tin rất phức tạp. Nếu các agent phải chịu trách nhiệm thiết kế cho việc phục hồi những bản tin TRAP thì càng làm tăng thêm độ phức tạp trong việc quản lý các agent trong môi trường đa nhà chế tạo. Người ta cũng đã cố gắng cải tiến cơ chế xử lý bản tin sự cố cho phiên bản thứ hai của SNMP. Thứ nhất là đơn nguyên TRAP được bỏ đi và thay thế nó bằng một lệnh GET/RESPONSE. Lệnh này do agent tạo ra và chuyển đến cho “manager bẫy” tại cổng UDP-162. Điều này phản ánh quan điểm là bộ phận quản lý sự cố có thể thống nhất các bản tin sự cố rồi trả lời cho các yêu cầu ảo. Bằng cách bỏ đi một đơn thể, giao thức được đơn giản hóa. Người ta cũng bổ sung thêm một cơ sở thông tin quản lí đặc biệt TRAP MIB để thống nhất việc xử lý sự cố, các manager nhận bản tin về các sự cố này và việc lặp lại được thực hiện để cải thiện độ tin cậy trong việc vận chuyển thông tin.

*Ảnh hưởng của tầng vận chuyển tới khả năng quản lí mạng:* Việc sử dụng mạng bị quản lý để hỗ trợ các nhu cầu thông tin liên lạc quản lý (quản lý trong băng) đã gây ra nhiều vấn đề thú vị. Việc quản lý trong băng và ngoài băng độc lập với việc lựa chọn giao thức quản lý. Quản lý trong băng có thể dẫn đến tình trạng mất liên lạc với một agent đúng lúc agent đó cần sự chú ý về quản lý (tùy thuộc vào nguồn của sự cố). Người ta có thể làm giảm nhẹ được vấn đề này nếu chính các thực thể mà agent quản lý lại bảo vệ đường truy nhập tới các agent này. Có một ảnh hưởng nhỏ về khả năng quản lý xuất hiện trong việc đánh địa chỉ tầng vận chuyển. Ví dụ: có thể xác định duy nhất một agent SNMP bằng địa chỉ IP và số cổng UDP. Điều này có nghĩa là với một địa chỉ IP cho trước thì ta chỉ có thể tiếp cận được một agent duy nhất. Hơn thế nữa agent này lại chỉ duy trì một cơ sở thông tin quản lí MIB duy nhất. Do vậy, với một địa chỉ IP duy nhất chỉ tồn tại một MIB. Việc gắn kết MIB với địa chỉ IP có thể hạn chế được độ phức tạp của biến số liệu mà agent cung cấp. Xem xét trong cùng một hoàn cảnh trong đó hệ thống yêu cầu nhiều MIB để quản lý các thành phần khác nhau của nó. Cần phải thống nhất các MIB khác nhau này dưới một cây MIB tĩnh duy nhất để có thể truy nhập chúng thông qua một agent duy nhất. Trong một số hoàn cảnh nhất định, việc thống nhất đó không thể thực hiện được. Trong những trường hợp như vậy, mỗi MIB đòi hỏi phải có riêng một nhóm giao thức SNMP/UDP/IP. Điều này làm tăng phức tạp trong việc tổ chức quản lý (các thông tin tương quan từ nhiều MIB thuộc một hệ thống cho trước) cũng như việc truy nhập nó (thông qua nhiều địa chỉ IP). Có một cách khác là một agent duy nhất trong một hệ thống có thể giữ vai trò như một proxy mở rộng cho các agent phụ đóng gói những cơ sở dữ liệu MIB khác nhau cùng liên quan tới một phân hệ cho trước. Các phiên bản mở rộng SNMPv2 hỗ trợ phương pháp này để xử lý nhu cầu truyền thông của manager. Các phiên bản mở rộng này cho phép agent đóng vai trò như một manager của các agent con tại chỗ, do vậy cho phép tiếp cận hàng loạt các agent con.

### 2.5.Các cơ chế bảo mật SNMP

Một SNMP management station có thể quản lý/giám sát nhiều SNMP element, thông qua hoạt động gửi request và nhận trap. Tuy nhiên một SNMP element có thể được cấu hình để chỉ cho phép các SNMP management station nào đó được phép quản lý/giám sát mình. Các cơ chế bảo mật đơn giản này gồm có: community string, view và SNMP access control list.

*Community string:* Community string là một chuỗi ký tự được cài đặt giống nhau trên cả SNMP manager và SNMP agent,đóng vai trò như “mật khẩu” giữa 2 bên khi trao đổi dữ liệu. Community string có 3 loại: Read-community,Write-Community và Trap-Community.Khi manager gửi GetRequest, GetNextRequest đến agent thì trong bản tin gửi đi có chứa Read-Community. Khi agent nhận được bản tin request thì nó sẽ so sánh Read-community do manager gửi và Read-community mà nó được cài đặt. Nếu 2 chuỗi này giống nhau, agent sẽ trả lời; nếu 2 chuỗi này khác nhau, agent sẽ không trả lời.

*View:* Khi manager có read-community thì nó có thể đọc toàn bộ OID của agent. Tuy nhiên agent có thể quyđịnh chỉ cho phép đọc một số OID có liên quan nhau, tức là chỉ đọc được một phần của MIB. Tập con củaMIB này gọi là view, trên agent có thể định nghĩa nhiều view.

*SNMP access control list:* SNMP ACL là một danh sách các địa chỉ IP được phép quản lý/giám sát agent, nó chỉ áp dụng riêng chogiao thức SNMP và được cài trên agent. Nếu một manager có IP không được phép trong ACL gửi request thìagent sẽ không xử lý, dù request có community string là đúng.Đa số các thiết bị tương thích SNMP đều cho phép thiết lập SNMP ACL.

# II.Hệ thống giám sát và quản lý mạngCacti

## 1.Các hệ thống giám sát và quản lý mạng

### 1.1.Tổng quan

NMS (Network Monitoring System) là một bộ phần mềm được thiết kế nhằm cải thiện hiệu quả và năng suất của quản lý mạng. Hầu hết các giải pháp NMS đều cung cấp 3 chức năng chính: giám sát, điều khiển và báo cáo.

Một số hệ thống NMS tiêu biểu được đánh giá cao: Cacti, Nagio, Open NMS, Network Monitor ...

### 1.2.Cấu trúc của một hệ quản lý mạng

Một vài điểm mà một NMS cần có:

* Hệ thống phải cung cấp một giao diện đồ họa mà tại đó nó có thể đưa ra hình ảnh của mạng theo từng cấp và nối kết logic giữa các hệ thống, nó cần giải thích rõ ràng các nối kết trong biểu đồ phân cấp chức năng và quan hệ của chúng như thế nào hiệu quả của mạng. Một giao diện đồ họa phải trùng với cấu trúc phân cấp chức năng. Một bản đồ mạng phải cung cấp chính xác hình trạng mạng.
* Hệ thống phải cung cấp một CSDL, CSDL này có khả năng lưu giữ và cung cấp bất kì thông tin nào liên quan đến hoạt động và sử dụng mạng, đặc biệt có thể quản lý cấu hình và quản lý tài khoản một cách hiệu quả.
* Hệ thống phải cung cấp một phương tiện thu thập thông tin từ tất cả các thiết bị mạng. Trường hợp lý tưởng cho người dùng là thông qua một giao thức quản lý mạng đơn giản.
* Hệ thống phải dễ dàng mở rộng và nâng cấp, cũng như thay đổi theo yêu cầu. Hệ thống phải dễ dàng khi thêm vào các ứng dụng và các đặc điểm yêu cầu của người kỹ sư mạng.
* Hệ thống phải có khả năng theo dõi các vấn đề phát sinh hoặc hậu quả từ bên ngoài. Khi kích cỡ và độ phức tạp của mạng tăng lên thì ứng dụng này trở lên

vô giá.

### 1.3.Các chức năng hệ thống NMS

*Quản lý hiệu năng:*

* Giám sát trạng thái theo thời gian thực ( CPU, Ram, Disk, device up/down, connected/lost connection, bandwith ...)
* Đưa ra kết quả thống kê ( sent/ received bytes, number active/ connected clients, resource usage ...)
* Hourly, daily, weekly reporting
* Top traffic usage ...

*Quản lý sự kiện:*

* Thu thập sự kiện theo thời gian thực, tổng hợp, phân loại lỗi theo mức độ nghiêm trọng.
* Tra cứu các sự kiện lỗi đã xảy ra.
* Hiển thị lỗi trên bản đồ mạng.
* Cảnh báo lỗi với người có trách nhiệm qua SMS, email, hoặc tới hệ thống như trouble ticket.

*Network Topology, Auto Discovery, Network Planning:*

* Mô hình hóa mô hình hệ thống giám sát, tự động giám sát và phát hiện các thay đổi trong mạng và cập nhật. Hỗ trợ Network Planning giúp tối ưu mạng giám sát.

*Tích hợp thông tin trên bản đồ mạng:*

* Hiển thị vị trí và loại thiế bị trên bản đồ.
* Quản lý thông tin hiệu năng cơ bản trên bản đồ.
* Định vị thiết bị lỗi qua bản đồ.
* Nhóm các thiết bị theo khu vực quản lý (địa lý).

*Quản lý thông tin Inventory:*

* Lưu trữ thông tin triển khai: vị trí, loại thiết bị, phiên bản phần mềm, số lượng vật tư đã dùng.
* Thống kê theo các loại thiết bị.

*Phân quyền người dùng hệ thống:*

Chức năng phân quyền linh hoạt, bảo mật.

* Có thể thiết lập quyền cho từng người dùng.
* Bổ sung hoặc xóa quyền người dùng khi cần thiết.
* Có thẻ định nghĩa và set quyền cho loại người dùng mới.

## 2.Cacti

### 2.1.Giới thiệu

Cacti được viết bởi Ian Berry và phiên bản đầu tiên (v0.6) được ra đời ngày 21/11/2011. Sau đó Cacti đã được nhiều người biết đến trong cộng đồng mã nguồn mở và ngày càng được phát triển và hoàn thiện qua từng phiên bản.

Cacti là một công cụ mã nguồn mở giám sát mạng và tạo biểu đồ được viết dựa trên ngôn ngữ chính là PHP và PERL, CSDL chính là MySQL và hoạt động trên giao diện web. Nó sử dụng RRDTool engine để lưu trữ dữ liệu, tạo đồ thị và thu thập các dữ liệu định kỳ ( tải hệ thống, tình trạng liên kết mạng, không gian đĩa cứng, đăng nhập người dùng... ) thông qua NET-SNMP.

Cacti là một chương trình đồ họa cho thống kê mạng, nó được thiết kế để sử dụng dễ dàng cho những ai lần đầu làm việc với hệ thống. Cacti tương đối dễ sử dụng cho các quản trị viên, những người mới chưa có nhiều kinh nghiệm.

### 2.2.Kiến trúc và hoạt động của Cacti.

**a.Kiến trúc**



*Hình 5. Sơ đồ khối kiến trúc Cacti*

Người dùng (thường là các quản trị viên) làm việc với Cacti thông qua các trình duyệt web (Browser). Với trình duyệt web, người quản trị khai báo các loại thiết bị trong mạng, thiết lập các thông số về thiết bị cần giám sát, quản lý, khai báo các phương thức nhận, truyền dữ liệu, đưa ra các yêu cầu về dạng sơ dồ hiển thị kết quả.

Những thông tin đăng nhập, thông số khai báo, thông tin thiết lập... được đưa vào từ người dùng sẽ được Cacti lưu lại trong cơ sở dữ liệu MySQL để duy trì hoạt động ở những lần sau đó.

Từ yêu cầu người dùng, Cacti sẽ xử lý các dữ liệu từ Poller. Poller liên tục lấy dữ liệu từ các thiết bị server, router, các ứng dụng ... (Các dữ liệu thu thập được sẽ lưu trữ bằng các cơ sở dữ liệu RRD). Cuối cùng Cacti sẽ hiển thị kết quả dưới dạng các sơ đồ:



*Hình 6. Hình ảnh dạng đồ thị kết quả của Cacti*

Trên thực tế các thành phần cài đặt trong hệ quản trị Cacti là các gói phần mềm: Webserver, PHP, RRDTool, MySQL, NET-SNMP.

* *Web server*: Hầu hết Cacti được xây dựng trên giao diện web nên web server là thành phần không thể thiếu. Bất kỳ web server hỗ trợ PHP nào cũng có thể dùng để cài đặt để Cacti giao tiếp nhưng Http của Apache hay IIS của Microsoft thì thường được khuyên dùng hơn cả bởi tính ổn định và phổ biến.
* *PHP*: Cacti được lập trình bởi ngôn ngữ PHP, do đó PHP cài phải được thiết lập trên hệ thống.
* *RRDTool*: Công cụ để lưu lại dữ liệu thu được và hỗ trợ việc xuất dữ liệu đồ họa.
* *MySQL*: Cacti lưu lại tất cả dữ liệu, history, dữ liệu về người dùng, mật khẩu...vào MySQL. MySQL được cài đặt giống như cơ sở dữ liệu riêng của Cacti để Cacti có thể tùy ý sử dụng. MySQL có thể cài đặt trên một máy chủ khác Cacti.
* *NET-SNMP*: Gói phần mềm hỗ trợ việc sử dụng và thực thi giao thức SNMP (v1,v2 và v3) có thể hoạt động trên Ipv4, Ipv6, IPX, AAL5..

**b.RRDTool**

RRDTool ( Round Robin Database Tool ) là một hệ thống ghi và vẽ dự liệu hiệu năng cao, được thiết kế để xử lý dữ liệu theo chuỗi thời gian, như là băng thông, nhiệt đồ phòng, CPU load, sever load, và để giám sát các thiết bị như là router, UPS, ... là một giải pháp mã nguồn mở. Nó cho phép người quản trị ghi và phân tích dữ liệu thu thập được từ tất cả các nguồn dữ liệu. Việc phân tích dữ liệu của RRDTool phụ thuộc vào khả năng tạo ra các đồ thị hiển thị các giá trị thu thập được trong một khoảng thời gian định kỳ.

RRDTool được viết bởi ngôn ngữ C và lưu trữ dữ liệu của nó trong file .rrd. Số lượng bản ghi trong một file .rrd đơn không bao giờ tăng, nghĩa là các bản ghi cũ sẽ thường xuyên được loại bỏ.

Các bước trong quá trình hoạt động của RRDTool:

* *Thu thập dữ liệu*: Dữ liệu thu thập được trong một khoảng thời gian cố định sẽ được lưu trong một cơ sở dữ liệu vòng.
* *Hợp nhất dữ liệu*: Việc quản lỹ dữ liệu của nhiều thiết bị trong một khoảng thời gian lớn (ví dụ một tháng) sẽ gây khó khăn cho người quản trị. RRDTool cung cấp cho người quản trị một số hoàm chức năng: xác định max, min, average của dữ liệu.
* *Lưu trữ vòng dữ liệu hợp nhất*.
* *Dữ liệu chưa biết*: Khi dữ liệu không tồn tại do thiết bị hỏng hoặc nguyên nhân khác, RRDTool sẽ lưu trữ file .rrd với giá trị dữ liệu \*UNKNOWN\*
* *Vẽ đồ thị*: Cho phép người quản trị tạo ra các báo cáo ở dạng đồ thị dựa trên dữ liệu được lưu trữ trong cơ sở dữ liệu.

**c.Hoạt động của Cacti**

Hoạt động của Cacti gồm 3 tác vụ:

* *Truy hồi dữ liệu (data retrieval):* Cacti truy hồi dữ liệu thông qua poller và sử dụng SNMP để truy hồi dữ liệu các thiết bị từ xa (các thiết bị này phải hỗ trợ SNMP).
* *Lưu trữ dữ liệu (data storage):* sử dụng RRDTool để lưu trữ dữ liệu.
* *Hiển thị dữ liệu (data presentation):* Cacti sẽ sử dụng chức năng tạo đồ thị để thực thi báo cáo dựa trên dữ liệu chuỗi thời gian thu thập được từ các thiết bị ở xa.

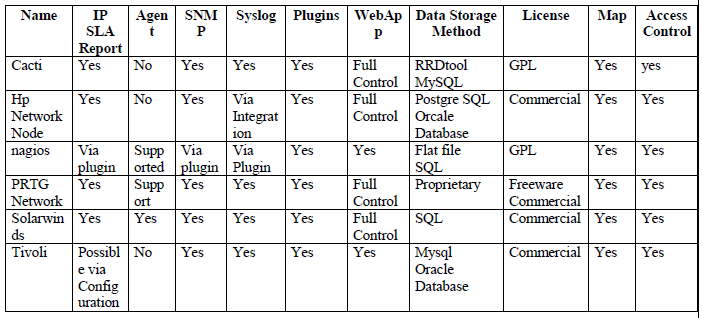
Hiện nay mạng lưới cơ sở hạ tầng chứ rất nhiều cac thiế bị khác nhau như router, switch, máy chủ, UPS, và các máy tính khác nhau và các thiết bị mạng. Để lấy dữ liệu từ các thiết bị này từ xa, Cacti sử dụng SNMP.

Những thiết bị có khả năng sử dụng SNMP đều có thể được theo dõi bởi Cacti. Cacti cho phép người quản trị có thể thêm nhiều plugin, tác động trực tiếp lên phần mềm để công việc theo dõi phù hợp với hệ thống và thuận lợi nhất ( có thể thêm plugin monitor, manage, weathermap, thold ...) để có thể đưa ra cảnh báo kịp thời khi có sự cố, nhằm ngăn chặn và xử lý nhanh, tốt nhất để hệ thống luôn hoạt động hiệu quả. Bằng nhiều cách như gửi thông báo qua email, báo động trực tiếp bằng âm thanh, mô hình hóa hệ thông sao cho dễ quản lý nhất...

### 2.3. Các chức năng chính của Cacti

* *Quản lý thiết bị*: Cacti giám sát và cho biết trạng thái của các thiết bị trong mạng theo thời gian thực. Cacti còn có thể thống kê hiệu quả hoạt động của các thiết bị.
* *Giám sát Traffic mạng:* Cacti giám sát hoạt động traffic trong hệ thống mạng, giám sát traffic trên từng cổng thiết bị.
* *Discovery*: Cacti có khả năng phát hiện các thiết bị mới kết nối vào hệ thống, kiểm tra xem thiết bị mới có hỗ trợ SNMP hay không, tạo điều kiện để kết nối giám sát thiết bị.
* *Chức năng cảnh báo âm thanh*: Khi sử dụng plugin monitor, hệ thông sẽ phát ra âm thanh cảnh bảo khi có thiết bị mất kết nối.
* *Chức năng cản báo qua email Thold*: Thold là một plugin cho phép thiết lập các hạng ngạch giám sát, và cảnh báo qua email. Thold thiết lập các giới hạn cao thấp của các chỉ số của thiết bị như CPU, traffic... nếu các chỉ số vượt giới hạn thì thold sẽ gửi email cảnh báo.
* *Chức năng router config*: Routerconfigs là plugin cho phép lưu lại thông tin cấu hình của router, nhờ đó có thể backup lại router khi cầ thiết.
* *Chức năng weathermap*: Weathermap là plugin cho phép mô phỏng lại hệ thống bằng biểu đồ có kèm các thông số theo dõi. Đây là chức năng rất trực quan và hữu ích.
* *Chức năng plugin NPC*: NPC là plugin giúp tích hợp Nagios vào Cacti, nhờ dố Cacti có thêm khả năng giám sát mạng của Nagios như: giám sát các dịch vụ mạng (pop3, smtp, ftp, http ...), giám sát thiết bị mạng, vẽ biểu đồ .

So sánh một vài giải pháp:



*Hình 7. So sánh Cacti với các hệ thống NMS khác*

### 2.4.Kiến trúc plugin Cacti

Với phần mã nguồn sẵn có, Cacti có thể hoạt động và cho người quản trị khả năng giám sát, theo dõi hệ thống mạng ở những khía cạnh cơ bản. Nhằm phát triển và mở rộng hệ quản trị dựa trên mã nguồn mở này, các tác giả của cacti vừa phát triển cacti, vừa tạo cho chính bản thân cacti có khả năng tích hợp thêm các chức năng được phát triển bởi người sử dụng hoặc các bên thứ ba (plugin) bằng cách xây dựng một tập các file hỗ trợ (Plugin Architecture).

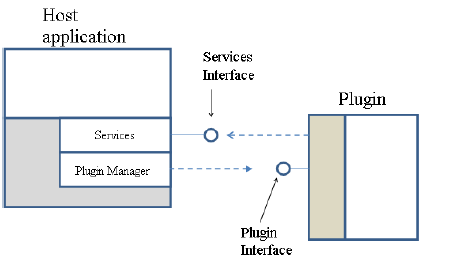
**a.Tổng quan plugin**

Plugin là một dạng phần mềm có thể thêm các tính năng vào một ứng dụng có sẵn. Ứng dụng có hỗ trợ Plugin là ứng dụng có khả năng tùy biến, thay đổi một cách linh hoạt. Ví dụ điển hình là các Plugin được sử dụng ở các trình duyệt web để thêm các tính năng: làm việc với các file có định dạng mở rông, công cụ tìm kiếm, diệt virus ...

Những lý do chính để ứng dụng cần tới Plugin:

* Cho phép nhà phát triển bên thứ ba có khả năng tạo mới những mở rộng cho ứng dụng.
* Hỗ trợ dễ dàng việc thêm vào các tính năng mới.
* Giảm kích cỡ của ứng dụng.
* Tách biệt phần mã nguồn với ứng dụng để tránh việc không tương thích license phần mềm.

Cơ chế hoạt động của plugin: Ứng dụng hỗ trợ Plugin cung cấp các dịch vụ mà Plugin có thể sử dụng (gồm cách mà plugin đăng kí, khai báo trong ứng dụng và giao thức trao đổi dữ liệu giữa ứng dụng và plugin). Các Plugin hoạt động phụ thuộc vào dịch vụ mà ứng dụng cung cấp chứ không tự khởi động và làm việc một cách riêng biệt được. Còn các ứng dụng hỗ trợ Plugin có thể hoạt động độc lập, giúp người sử dụng có thể làm việc với ứng dụng dù có hoặc không có bất lỳ Plugin nào.

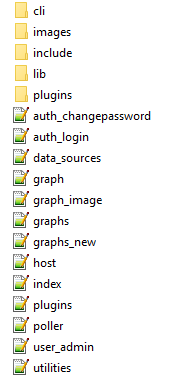


*Hình 8. Cơ chế hoạt động của plugin*

**b.Plugin Architecture trong hệ quản trị Cacti**

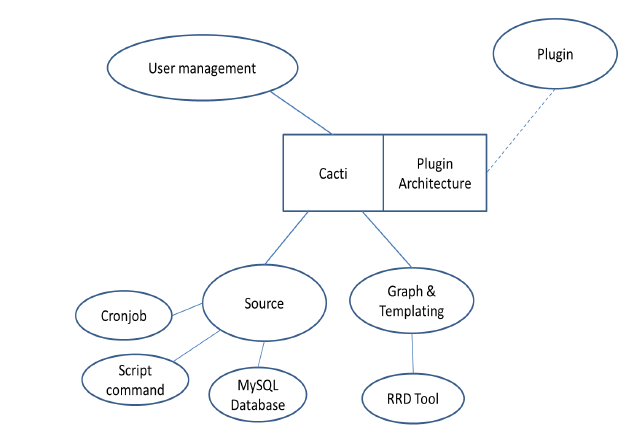
Bản thân Cacti chưa có khả năng hỗ trợ, tương tác với các Plugin. Vì vậy, các nhà phát triển thiết lập một tập các file nằm trong hệ quản trị mạng Cacti, nhằm giao tiếp với Plugin để có thể thực hiện tiếp mã nguồn mở hay thương mại hóa các tính năng thêm vào mà không cần thay đổi phần mã nguồn lõi của hệ quản trị này. Các bản vá lỗi bảo mật cho phần lõi cũng có thể được áp dụng một cách an toàn mà không xung đột với các chức năng của các plugin. Và từ đó cho phép người dùng cuối thực hiện tính năng còn thiếu hoặc tạo ra những cải tiến cụ thể cần thiết phù hợp với mục đích sử dụng của công việc như: thay đổi giao diện phù hợp để thân thiện hơn hat tập trung cho thương hiệu riêng, hỗ trợ truy cập vào CSDL của Cacti với những công cụ mở rộng, thực thi bộ đệm, cải tiến hiệu suất hỗ trợ triển khai hệ thống quản trị cỡ lớn, thêm các chức năng báo cáo, ghi thêm các sự kiện hệ thống cần thiết, lập ngường và cảnh báo ... cùng nhiều chức năng khác nữa.

Thành phần của Plugin Architecture bao gồm:



*Hình 9. Thành phần của Plugin Architecture*

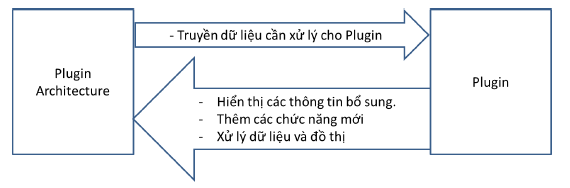
Tương tác giữa Plugin Architecture và các thành phần trong Cacti:



*Hình 10. Tương tác giữa Plugin Architecture và các thành phần trong Cacti*

Plugin Architecture được thiết lập và trở thành một phần của Cacti. Thành phần này có thể hoạt động để sử dụng các dữ liệu, tài nguyên của Cacti lấy từ Cronjob, Script command hay cơ sở dữ liệu MySQL , tuân thủ những quy định từ User Management và có thể thao tác với các chức năng đồ họa dựa trên RRD Tool như chính bản thân Cacti. Việc sử dụng tài nguyên và thao tác như Cacti, Plugin Architecture nhằm mục đích giao tiếp với các chương trình mở rộng, phát triển thêm các tính năng của Cacti, hay còn được gọi là Plugin.

Plugin Architecture giao tiếp với Plugin qua các Hook. Thông qua Hook, Plugin Architecture truyền các thông tin lấy được cho Plugin và Plugin có thể hiển thị thêm các thông tin lấy từ thiết bị, đưa vào các chức năng mới, làm việc với dữ liệu và đồ thị.

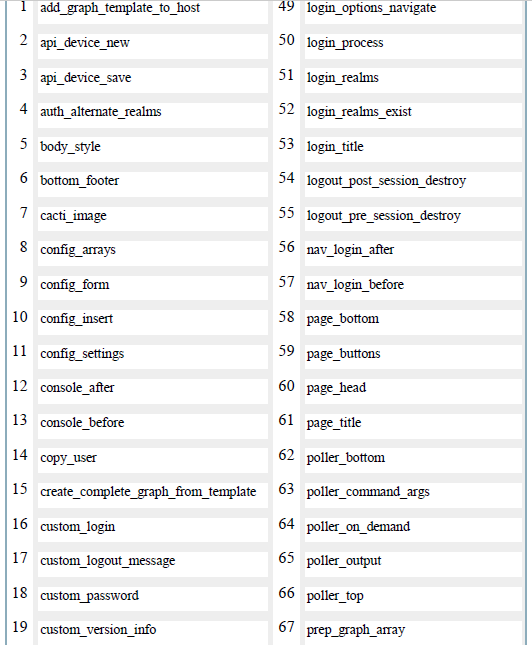


*Hình 11. Quan hệ giữa Plugin Architecture và Plugin*

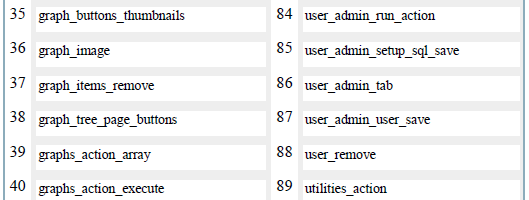
**c.Cacti Plugin**

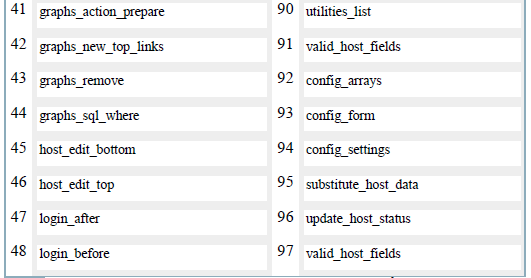
Cacti Plugin là các Plugin được thiết kế nhằm hỗ trợ cho Cacti, hoạt động trong môi trường Cacti. Mỗi Plugin có một cấu hình riêng ứng với mục đích, chức năng của Plugin. Nhưng cấu trúc chung đều gồm ít nhất 2 file: *index.php* và *setup.php*, ngoài ra có thể có thêm vài file khác chứa thông tin và hỗ trợ cho Plugin hoạt động (thông tin hướng dẫn, các thay đổi qua từng phiên bản, hình ảnh logo, các thư viện hỗ trợ, chức năng bổ sung).

Để giao tiếp với Plugin Architecture, Plugin có thể sử dụng các Hook nằm trong bảng dưới đây:









*Hình 12. Các Hook API*

**d.Khởi tạo Plugin**

Các Plugin hoạt động khi được kích hoạt, lấy dữ liệu từ tài nguyên của Cacti và xử lý sau đó trả kết quả thực hiện cho Cacti nên cần tuân theo những quy tắc sau:

*Với Cấu trúc thư mục*

Mỗi Plugin được đặt trong thư mục riêng, được đặt tên riêng phân biệt với các Plugin khác. Tốt nhất là nên để ở chữ thường, không nên đặt tên dài quá, chỉ chứa các ký tự trong bảng chữ cái abc và số thập phân.

Thư mục chứa Plugin đặt như sau:

/plugins/PLUGINNAME/

*Với file Setup*

Cacti sẽ tìm file setup.php trước tiên trong thư mục chứa Plugin. Trong file này chỉ nên chứa các hàm, không nên để mã lệnh có khả năng tự động thực thi. Ngoài các hàm thực thi các chức năng riêng, bắt buộc phải có các hàm sau:

* Hàm cài đặt Plugin: Đây là chức năng đầu tiên là chức năng cài đặt, kích hoạt khi có yêu cầu cài đặt từ Cacti. Không có tham số cho hàm này. Hàm này được khai báo như sau:

function plugin\_PLUGINNAME\_install ()

{ api\_plugin\_register\_hook('PLUGINNAME','top\_header\_tabs','PLUGINNAME\_show\_tab','setup.php');

}

* Hàm gỡ bỏ Plugin: Hàm được gọi khi có yêu cầu gỡ bỏ Plugin.

function\_PLUGINNAME\_uninstall () { // Các thao tác gỡ bỏ được đặt ở đây }

* Hàm phiên bản: Cho biết thông tin về phiên bản của Plugin và một vài thông tin khác. Ví dụ:

function plugin\_PLUGINNAME\_version () {

return array( 'name' => 'PLUGINNAME',

'version' => '1.0',

'longname' => 'Plugin Name',

'author' => 'Sinh vien',

'homepage' => 'http://soict.hust.edu.vn',

'email' => 'sinhvien@soict.hust.edu.vn',

'url' => 'http://soict.hust.edu.vn/' ); }

* Hàm kiểm tra cấu hình: nhằm xác định Plugin đã sẵn sàng để kích hoạt. Khi một Plugin được cài đặt nó bắt đầu ở trạng thái vô hiệu hóa (disable). Chỉ có vài thông số cần thiết được thiết lập. Hàm này sẽ cho biết các thiết lập là chính xác. Ví dụ:

function plugin\_PLUGINNAME\_check\_config () {

if (read\_config\_option('PLUGINNAME\_SETTING') != '')

{

return true;

}

return false; }

*Đăng ký sử dụng các Hook*

Để sử dụng Hook nhằm tương tác với Cacti, Plugin cần đăng ký mỗi Hook cho mỗi chức năng thực hiện. Việc này được thực hiện thông qua hàm:

api\_plugin\_register\_hook('PLUGINNAME','HOOKNAME','CALLBACKFUNCTION','FILENAME');

Trong đó:

PLUGINNAME –Tên của Plugin.

HOOKNAME –Tên của Hook cần đăng ký.

CALLBACKFUNCTION - Tên của hàm được gọi khi Hook kích hoạt.

FILENAME –Tên của tập tin chứa các chức năng trên.

*Đăng ký Realm*

Cacti chia thành các khu vực (Realms) khác nhau với quyên truy cập khác nhau. Để thực hiện trên các Realm, Plugin cần đăng ký quyên truy cập Realm (Realm Permission). Việc này được thực hiện thông qua hàm: api\_plugin\_register\_realm('PLUGINNAME','FILENAMETORESTRICT','DISPLAYTEXT',1);

Trong đó:

PLUGINNAME –Tên của Plugin.

FILENAMETORESTRICT –Tên tập tin hạn chế truy cập.

DISPLAYTEXT – Nội dung sẽ hiển thị trong User Permissions form của khu vực này.

Tham số cuối có thể là 1 hay 0 (true hoặc false). 1 nghĩa là gán quyền 'admin' cho user account (true).

# Phần B: PLUGIN GHI NHẬT KÝ

# I.Yêu cầu, ý tưởng và giải pháp

## 1.Yêu cầu

Hệ thống cho phép:

* Quản lý, giám sát các máy Client: Theo dõi các thông tin về CPU, ổ cứng, tốc độ mạng, process, loggin user ... và hiển thị dưới dạng đồ thị nếu thống kê trong khoảng thời gian và dưới dạng bảng biểu nếu thống kê tại một thời điểm.
* Gửi email cảnh báo đến người quản trị nếu các thông số theo dõi vượt ngưỡng cho phép.
* Ghi lại và hiển thị syslog và các sự kiện của hệ thống, của các máy clients.

## 2.Plugin ghi nhật ký

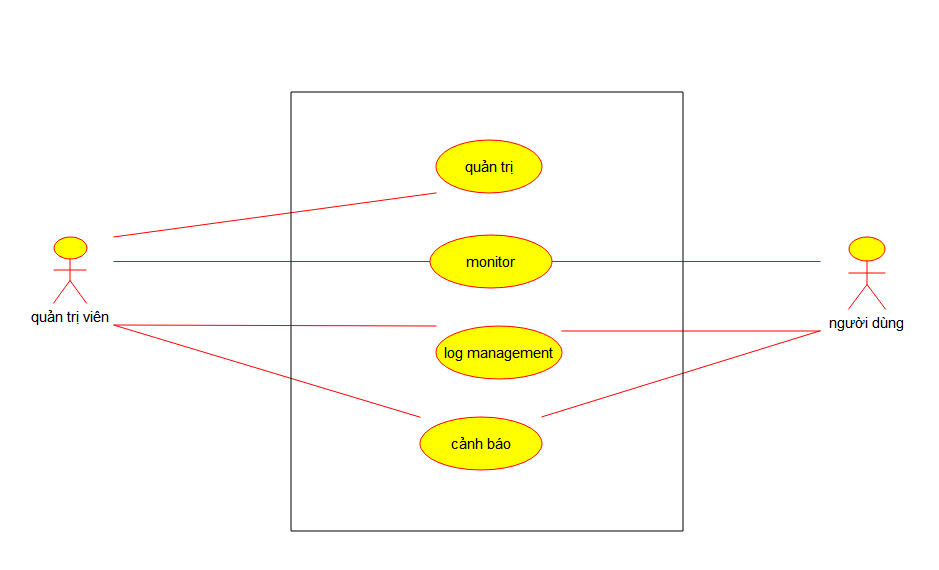
Sau quá trình tìm hiểu, xây dựng Plugin trên hệ thống NMS Cacti là một lựa chọn tối ưu.

Cacti được viết bằng ngôn ngữ PHP. Ta sử dụng thư viện hỗ trợ SNMP trên PHP là PHP-SNMP để có thể truy suất các thông tin của thiết bị, port trong hệ thống mạng, syslog hệ thống, bẫy các sự kiện từ các máy clients rồi lưu các thông tin vào cơ sở dữ liệu. Từ đó có thể thực hiện việc quản lý, giám sát và ghi lại syslog, event của các máy clients trong hệ thống mạng.

# II.Thiết kế hệ thống

## 1.Mô tả chương trình

Từ đặc tả yêu cầu của hệ thống, tiến hành phân tích và thiết kế hệ thống sẽ cần có các chức năng chính nhứ sau:

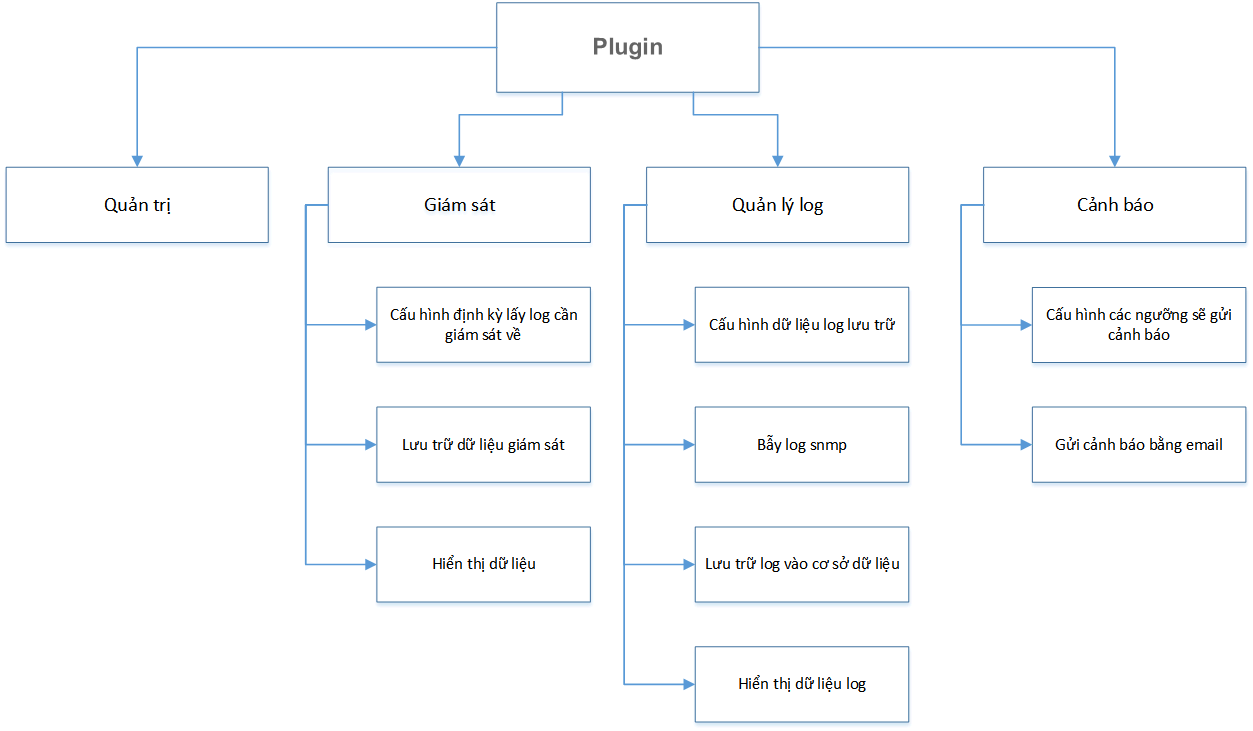


Chức năng của hệ thống

- **Chức năng quản trị**: quản trị viên có thể thay đổi các thông số và cấu hình qua giao diện

- **Chức năng giám sát**: theo dõi, thu thập số liệu của các thiết bị trong mạng trong mạng theo thời gian

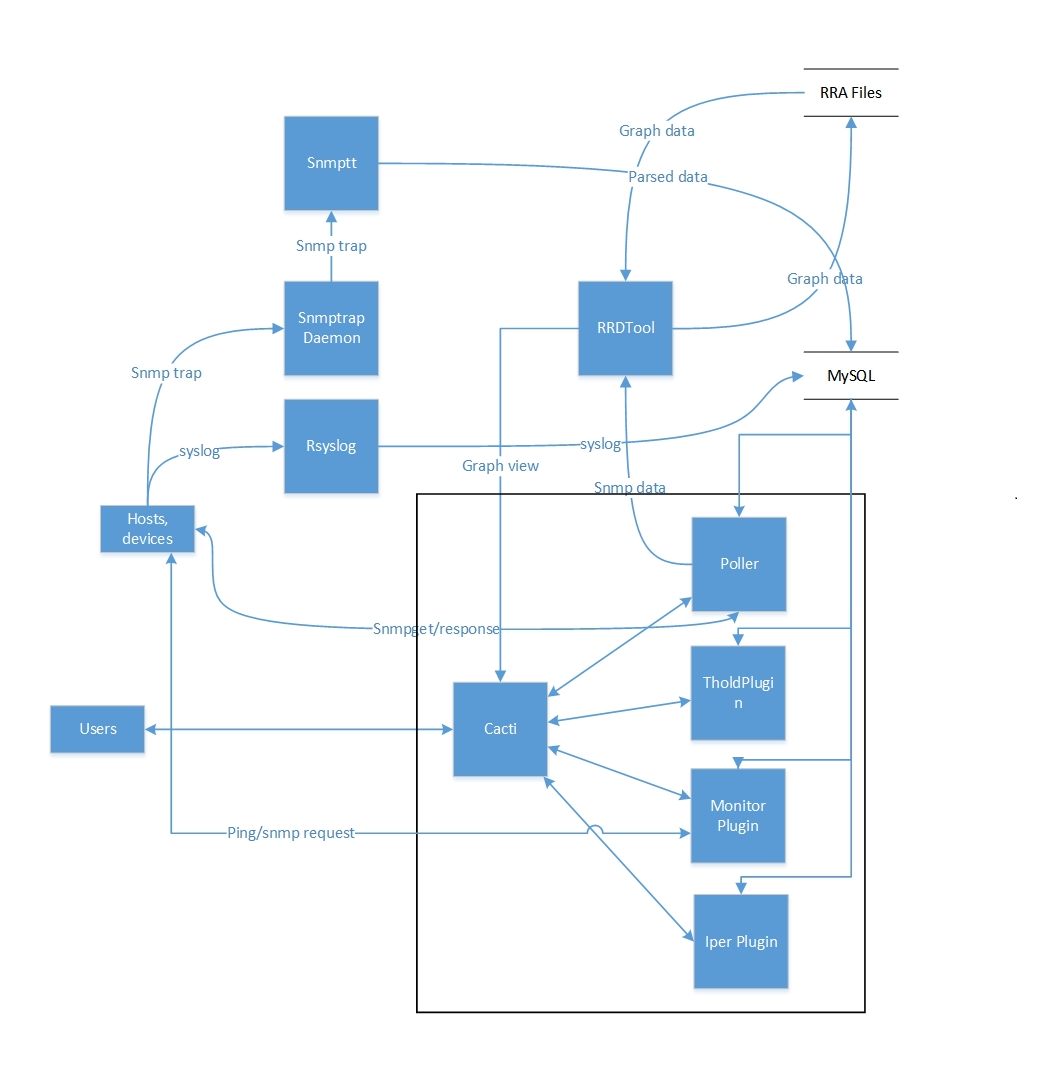
* Cho phép người quản trị cấu hình trên giao diện có thể lập lịch, định ký truy vấn thông tin phần cứng và phần mềm của các thiết bị
* Thông tin lấy về được lưu trữ trữ lại ra file và cơ sở dữ liệu của hệ thống
* Có thểm xem được các số liệu thu thập, sự thay đổi của số liệu theo thời gian qua giao diện hiển thị ở dạng bảng biểu và đồ thì
* **Quản lý log**: Các log của hệ thống, bẫy log (snmptrap) của các thiết bị trong mạng
  + Cho phép người quản trị cấu hình các thống số như thời gian lưu trữ, số bản ghi log tối đa lưu trong hệ thống,…
  + Lưu trữ các log của hệ thống vào trong cơ sở dữ liệu
  + Bẫy các log của các thiết bị trong mạng gửi về (snmp trap) lưu trữ vào cơ sở dữ liệu
  + Hiển thị log bắt được trên giao diện người dung
* **Chức năng cảnh báo**:
  + Cho phép quản trị cấu hình các các ngưỡng cảnh báo đến người dùng khi có các số liệu bất thường trong hệ thống (Thiết bị mất kết nối, tắt máy, hết bộ nhớ,…)
  + Chức năng gửi email cho người dùng

****

Mô hình phân rã chức năng

## 2.Mô tả hệ thống

Luồng dữ liệu và mô hình của hệ thống được thể hiện theo sơ đồ sau



- Cacti: là bộ khung của toàn hệ thống, là giáp hoàn thiện dự trên bộ công cụ RRDtool. Cacti cung cấp bộ chức năng, giao diện để người quản trị có thể thêm/bớt, tùy chỉnh các thiết bị cần giám sát. Các thành phần chính của cacti bao gồm

* Poller: thành phần truy vấn lấy thông tin của các agents cần theo dõi
* Data query, hiển thị: dữ liệu thu thập được và được lưu trữ vào các file rra của rrdtool sau đó hiển thị cho người dùng ở dạng biểu đồ

- Rrdtool: là một cơ sở dữ liệu ngắn hạn được sử dụng bởi cacti. Ứng dụng cung cấp chức năng lưu trữ dữ liệu định kỳ ra file và có thể truy vấn được dữ liệu trong các file này hiển thị ra ở dạng biểu đồ

- rra: là các file dữ liệu của rrdool

- MySQL: Lưu trữ các thông tin cấu hình chung của toàn hệ thống, các plugin, dữ liệu syslog, snmp log

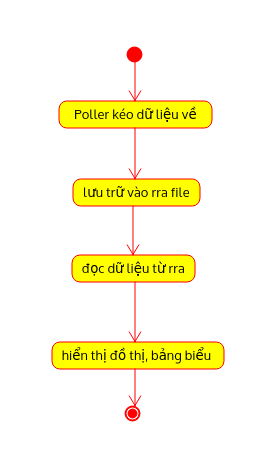
- Snmp daemon: là service chạy trên máy chủ cacti và các agent, chịu trách nhiệm phản hồi các truy vấn snmp gửi đến

- Snmp trap daemon: là service chạy trên máy chủ cacti chịu trách nhiệm xử lý các gói tin snmptrap của các agents gửi đến

- Rsyslog: là service chạy trên máy chủ cacti, chịu trách nhiệm xử lý các dữ liệu log của toàn hệ thống. Rsyslog có nhiều chức năng cho phép đẩy log ra nhiều luồng như ghi log vào file, ghi log vào cơ sở dữ liệu,…

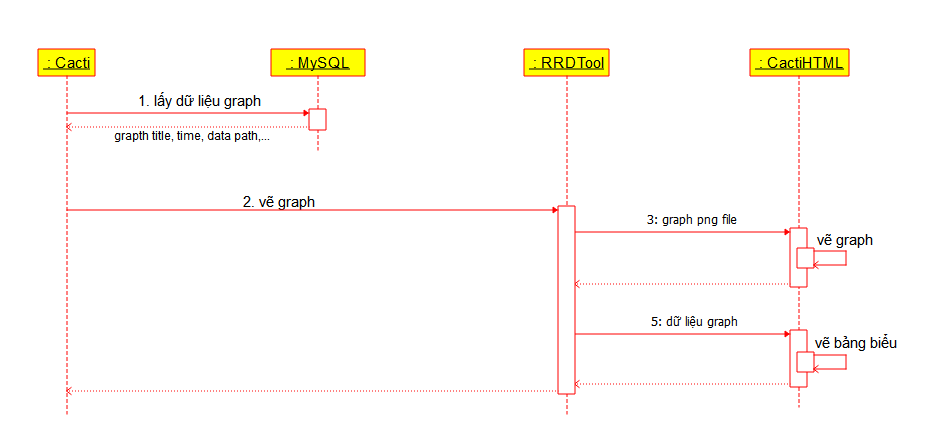
- Snmptt: là service chạy trên máy chủ cacti, chịu trách nhiệm bóc tách dữ liệu snmptrap sau đấy lưu kết quả ra syslog hoặc cơ sở dữ liệu.

-Luồng xử lý chức năng giám sát: Các loại dữ liệu cần giám sát (dữ liệu có thể là CPU, bộ nhớ, processes, card mạng…) sẽ được cấu hình và lưu trữ vào MySQL. Sau đấy, thành phần của poller sẽ định kì gửi các truy vấn snmp đến từng thiết bị để lấy các thông tin theo cấu hình (thông tin có thể là CPU, bộ nhớ, processes, card mạng…). Dữ liệu truy vấn được sẽ được RRDtool lưu trữ ra các file rra.



truy vấn và lưu trữ dữ liệu cần giám sát

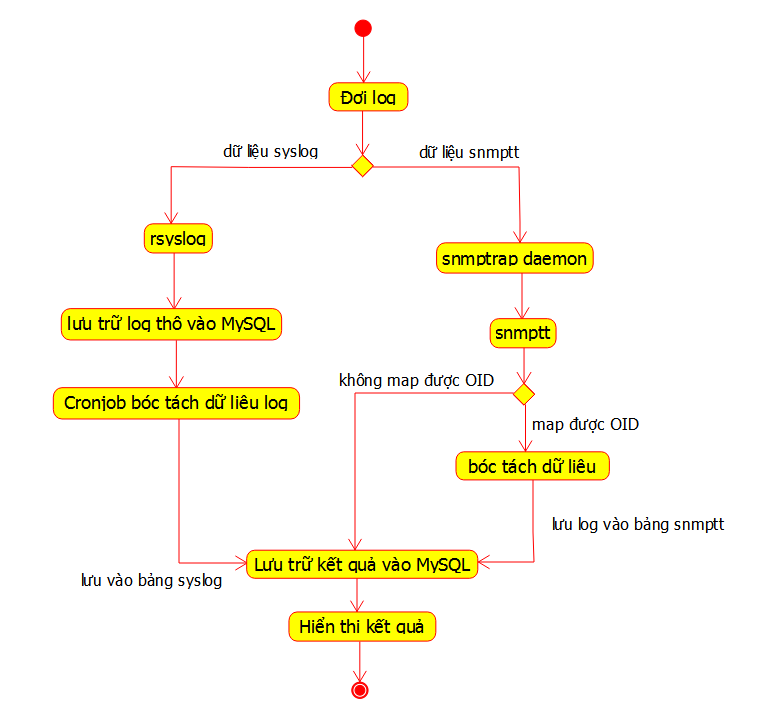
Khi người dùng sử dụng chức năng, dữ liệu sẽ được lấy ra từ cơ sở dữ liệu và từ rrdtool và hiển thị kết quả cho người dùng



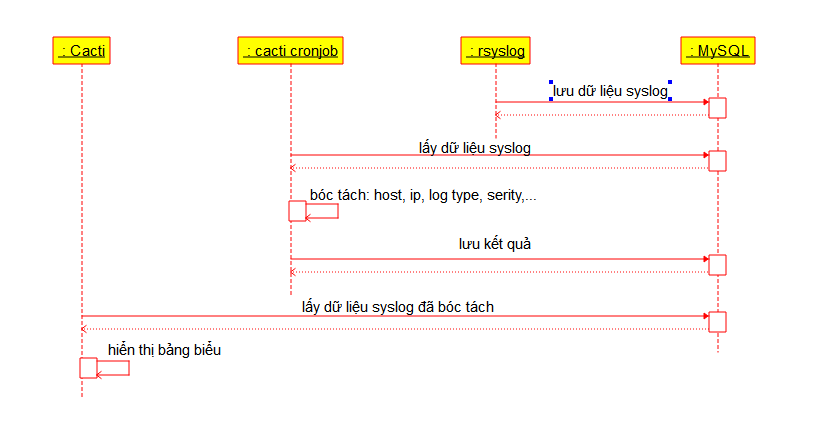
Hiển thị kết quả cho người dùng

-Luồng xử lý chức năng quản lý log:

Hệ thống quản lý hai loại dữ liệu log là syslog và snmp trap log. Dữ liệu syslog của hệ thống sẽ được xử lý bằng bộ công cụ rsyslog. Dữ liệu snmp trap được bắt bằng snmptrap daemon và bóc tách bằng bộ công cụ snmptt. Dữ liệu log sau khi xư lý sẽ được lưu trữ vào cơ sở dữ liệu để hiển thị cho người dùng.



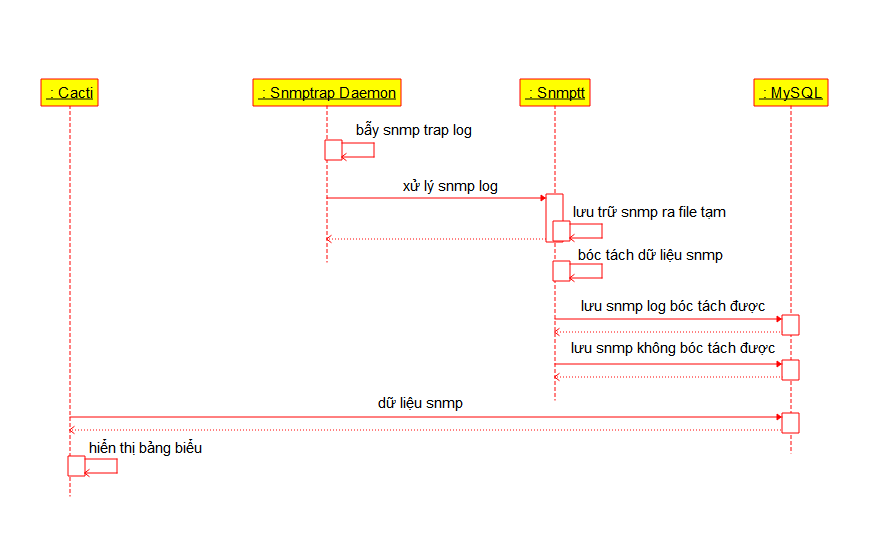
* Chức năng quản lý syslog: dữ liệu syslog của hệ thống sẽ được rsyslog lưu trữ tạm thời vào MySQL, dữ liệu này sau đó sẽ được Cacti lấy ra và tiến hành bóc tách nhận biết các thông tin như hostname, IP, loglevel, loại log, nội dung và lưu trữ kết quả vào cơ sở dữ liệu sau đấy sẽ hiển thị cho người dùng ở dạng bảng biểu



quản lý syslog

* Quản lý bẫy log snmp:

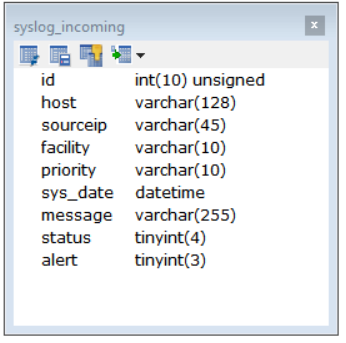
Snmptrap daemon sẽ đón các gói tin snmp trap được gửi về sau đó đẩy cho snmptt xử lý. Snmptt khi hoạt động ở chế độ daemon sẽ lưu trữ các log này ra file tạm. Sau đó định kỳ sẽ đọc dữ liệu này lên, tiến hành bóc tách để xác định các thông tin như hostname, IP, OID, OID name, kiểu dữ liệu, giá trị,… log bóc tách được và không bóc tách được đều được lưu trữ vào hai bảng khác nhau trong cơ sở dữ liệu. Dữ liệu này được hiển thị cho người dùng ở dạng bảng biểu



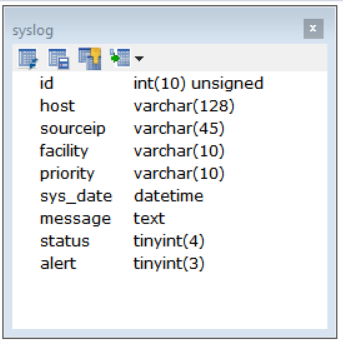
xử lý snmp trap

## 3.Thiết kế cơ sơ dữ liệu

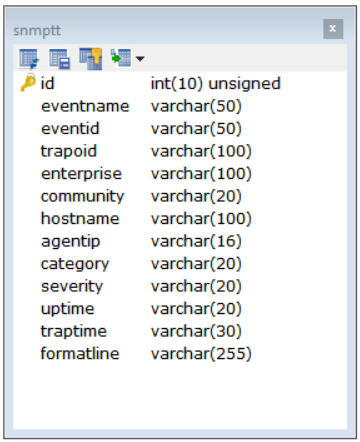
Bảng **syslog\_incoming** lưu trữ dữ liệu syslog thô chờ được bóc tách



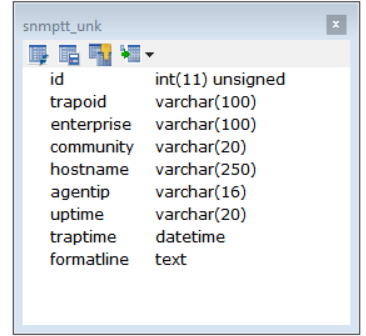
Bảng **syslog** lưu trữ dữ liệu syslog đã được xử lý xong



Bảng **snmptt** lưu trữ liệu snmp bẫy được và bóc tách được



Bảng **snmptt\_unk** lưu trữ dữ liệu snmp bẫy được nhưng không xử lý được



## 4.Các module và chức năng chính

- Monitor

- Camm

- THold

# Phần C: CÀI ĐẶT THỰC NGHIỆM VÀ ĐÁNH GIÁ

# I.Cài đặt và cấu hình

## 1.Cài đặt

### 1.1.Cài đặt Cacti

* Cacti 0.8.8f
* Ubuntu 14.04

Một số gói cần cài đặt trước khi cài đặt Cacti:

* RRDTool 1.0.49 or higher.
* NET-SNMP.
* MySQL 4.1.x or higher.
* PHP 4.3.6 or higher.
* Apache/ IIS or any other web server.

Cài đặt các gói sau bằng dòng lệnh trên terminal (Ctrl+Alt+T) Ubuntu:

*Cài đặt apache 2*:

sudo apt-get install apache2

*Cài đặt PHP:*

sudo apt-get install php5 libapache2-mod-php5

*Cài đặt MySQL:*

sudo apt-get install mysql-server-core-5.5 mysql-server-5.5. php5-mysql

Sau đó tạo password cho tài khoản root (ban đầu là trống).

*Cài đặt Net-SNMP:*

sudo apt-get install snmp php5-snmp

sudo apt-get install snmpd php5-snmp

*Cài đặt RRDTool:*

sudo apt-get install rrdtool

*Cài đặt rsyslog:*

sudo apt-get install rsyslog rsyslog-mysql

*Cài đặt snmptt:*

sudo apt-get install snmptt

*cài đặt các thư viện phụ thuộc cho snmptt:*

perl –MCPAN –e shell

install List::MoreUtils

install Cofig:IniFiles

install Bundle::DBI

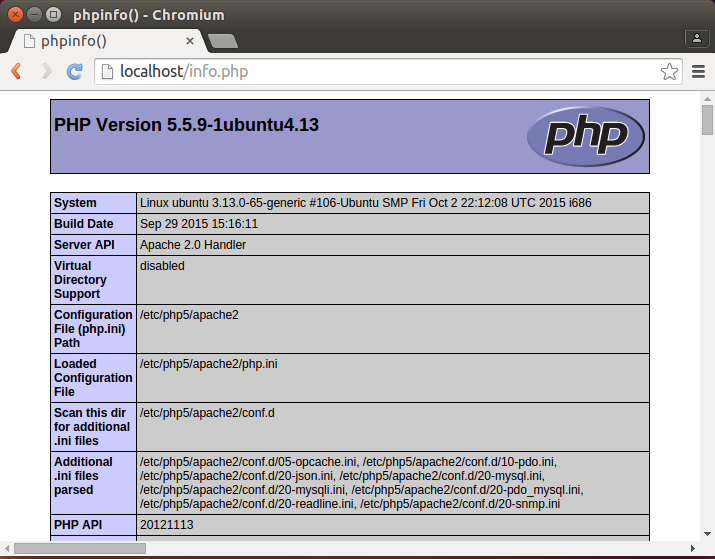
install DBD::mysql

Sau khi cài đặt, khởi động lại Apache2 bằng lệnh:

sudo /etc/init.d/apache2 restart

Sau cùng để kiểm tra kết quả đã cài đặt thành công hay chưa ta gõ vào thành URL địac chỉ: <http://localhost/info.php>

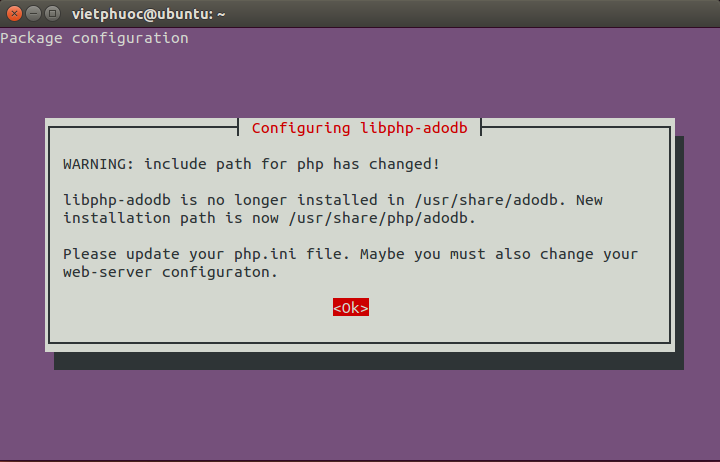
Kết quả:



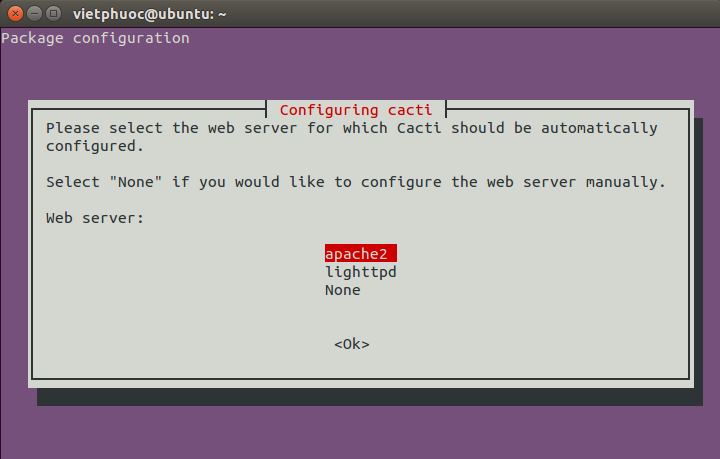
*Cài đặt Cacti:*

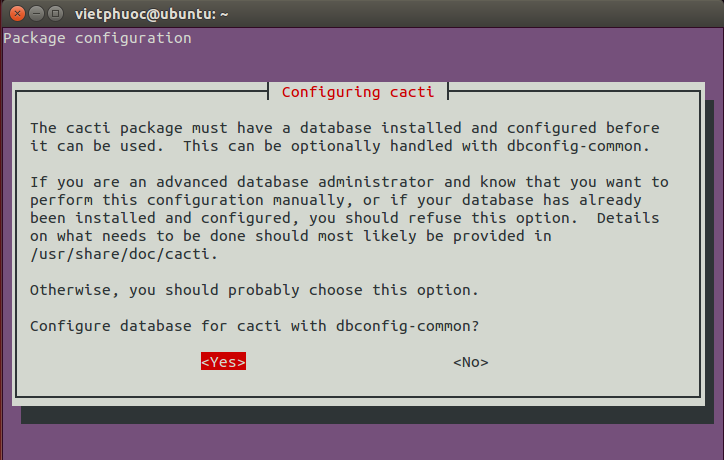
Cài đặt thành công các gói trên rồi tiến hành cài đặt Cacti:

sudo apt-get install cacti

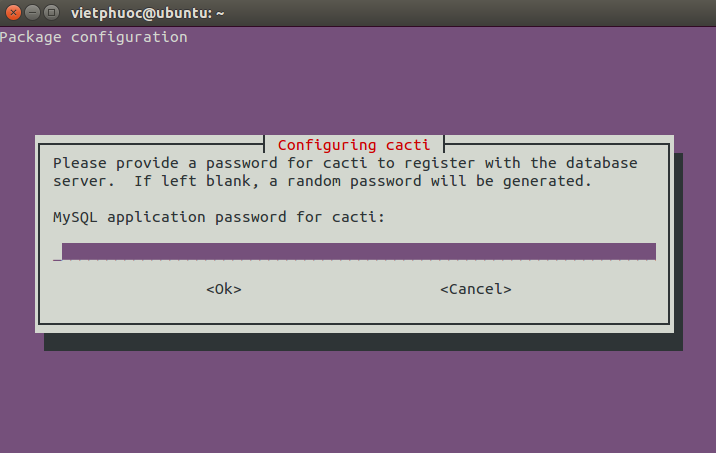


Chọn Web server là Apache2:



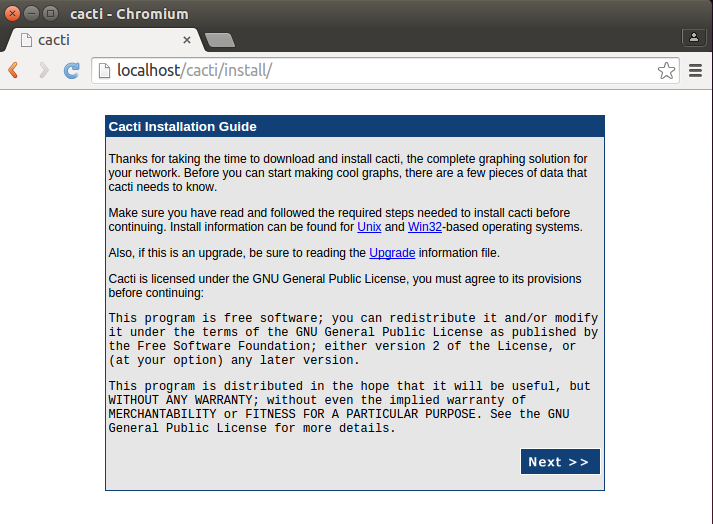


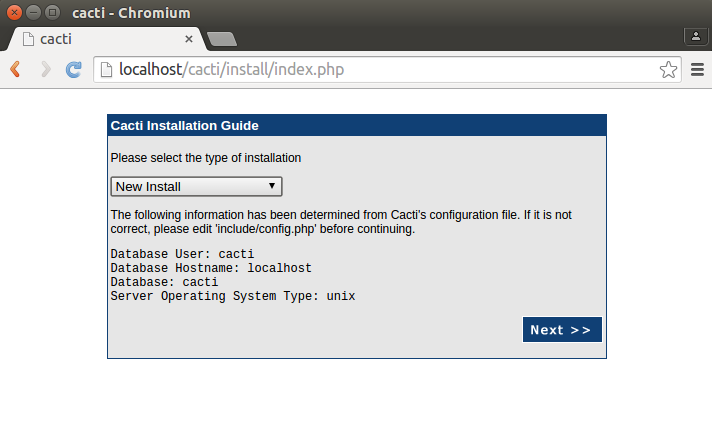
Đặt password cho database’s aministractive user và MySQL application password for cacti:



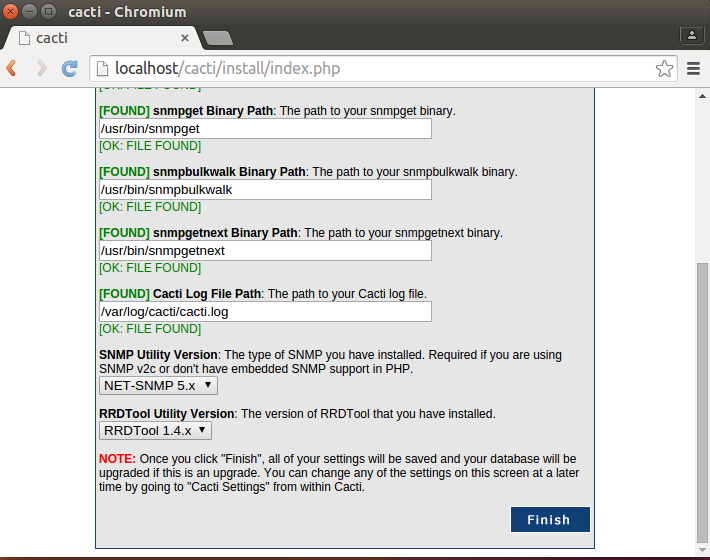
Hoàn tất việc cài đặt, gõ vào thanh URL địa chỉ: <http://localhost/cacti>

Kết quả:

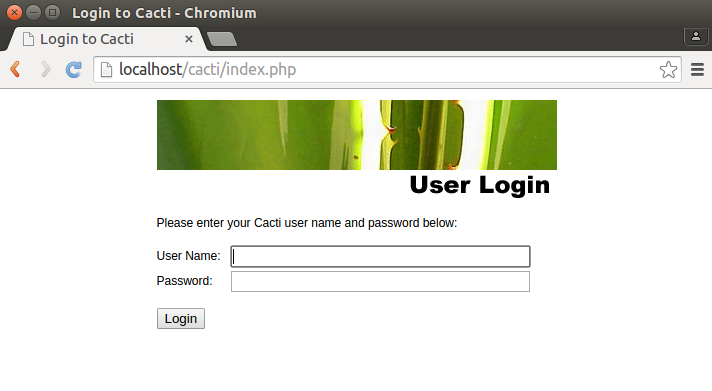




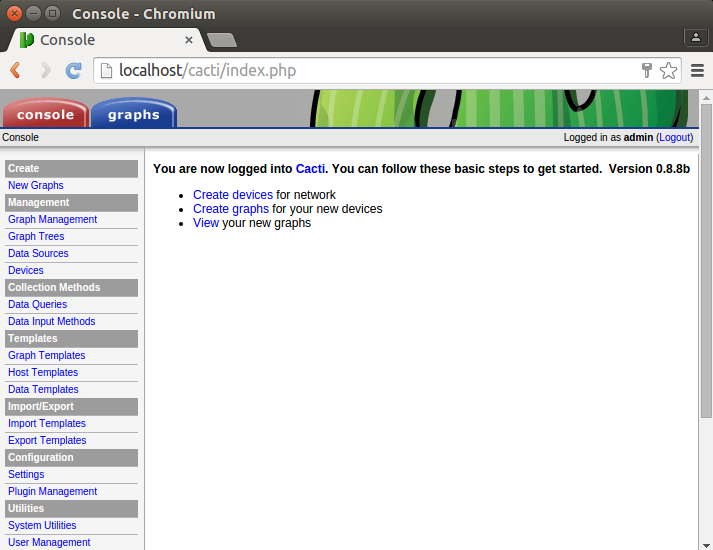
Ấn finish để hoàn tất cài đặt:



Màn hình đăng nhập:



Sau khi đăng nhập:



Việc cài đặt cacti đã thành công!

### 1.2.Cài đặt Plugin

Hệ thống dựa trên hai bộ plugin của cacti là monitor và camm đã chỉnh sửa lại. Việc cài đặt tương tự như các plugin chung của cacti.

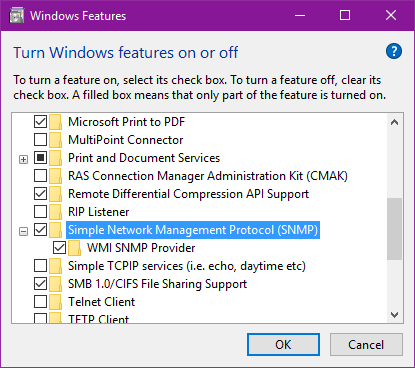
## 2.Cấu hình

### 2.1.Cấu hình trên Clients.

**a. Windows Client**

* Cài đặt dịch vụ SNMP

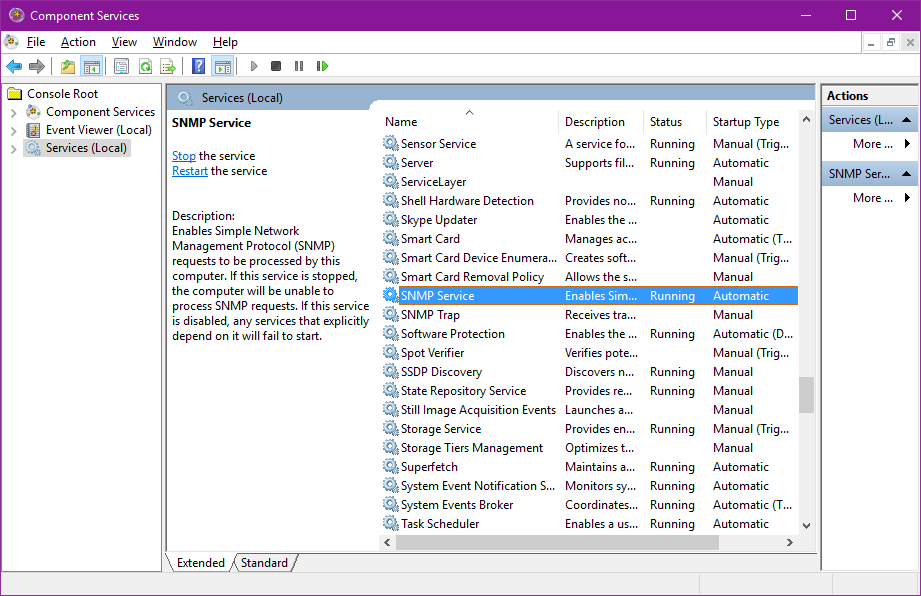
Control Panel -> Programs and Features -> Turn Windows features on or off -> chọn SNMP Service và SNMP WMI Provider.

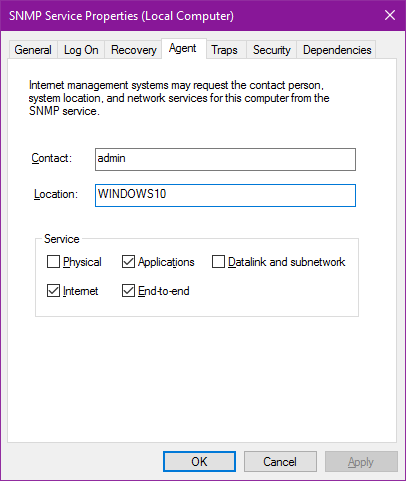
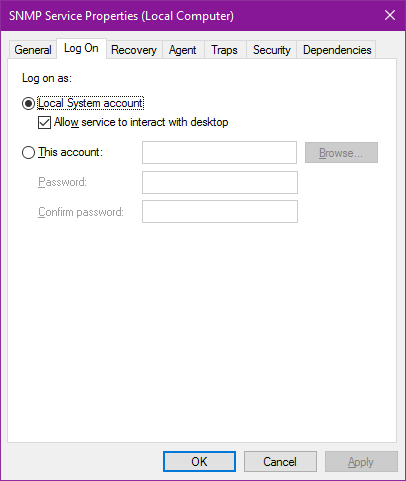


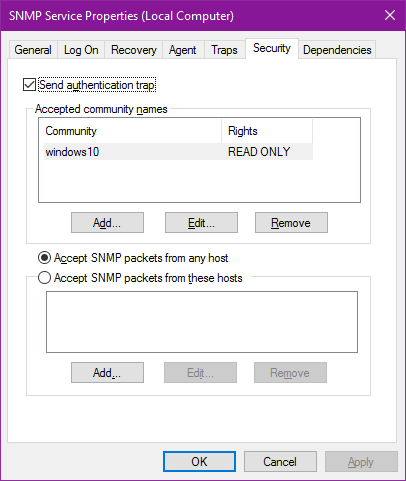
* Cấu hình SNMP Service

Control Panel -> Administrative Tools -> Services -> SNMP Service

Tiến hành cấu hình.







Restart SNMP service và SNMP trap.

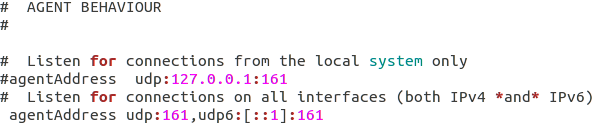
* Tắt Firewall trên máy.

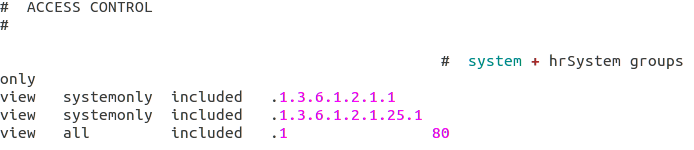
**b.Ubuntu Client**

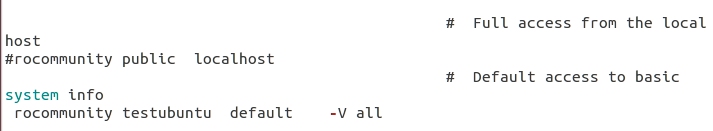
* Cài đặt SNMP

sudo ap-get install snmpd

* Cấu hình SNMP







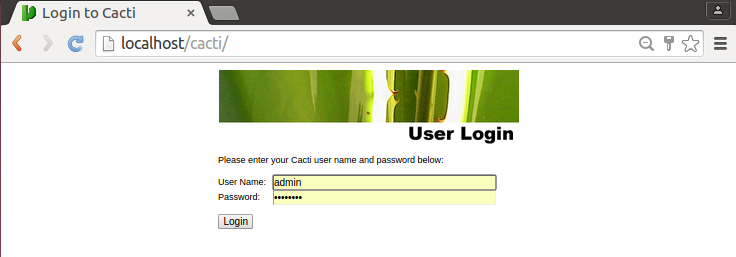
Sau đó restart SNMP service:

sudo snmpd service restart

### 2.2.Cấu hình trên localhost

**a. Add các Clients vào Cacti để quản lý**

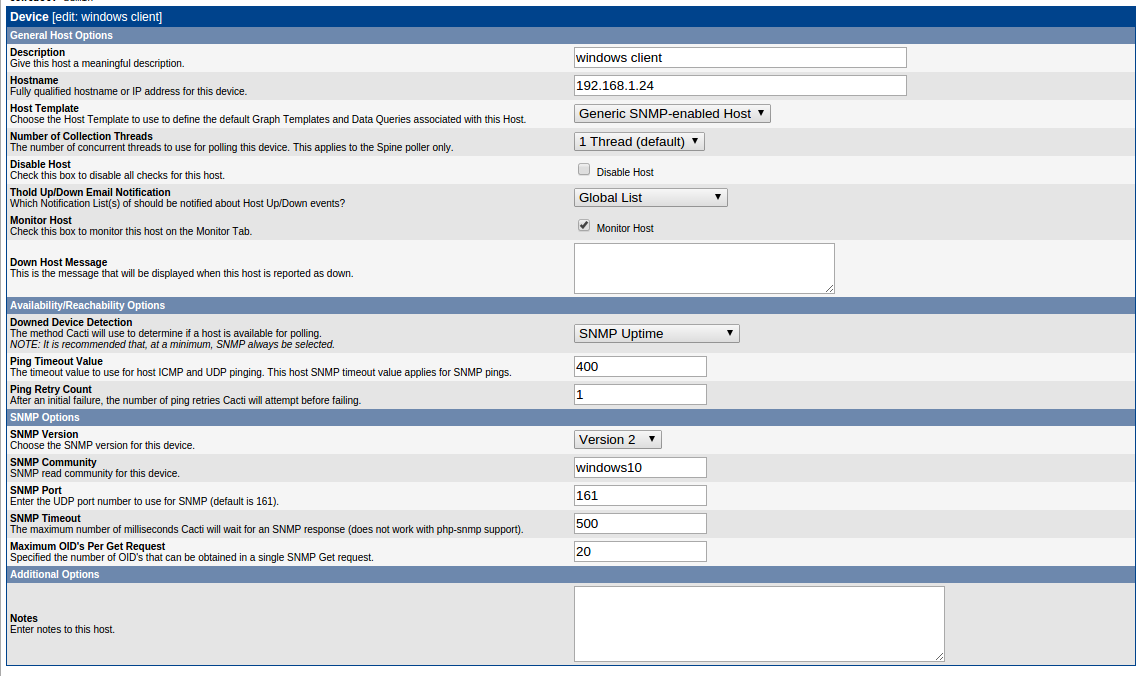
Đăng nhập vào Cacti:



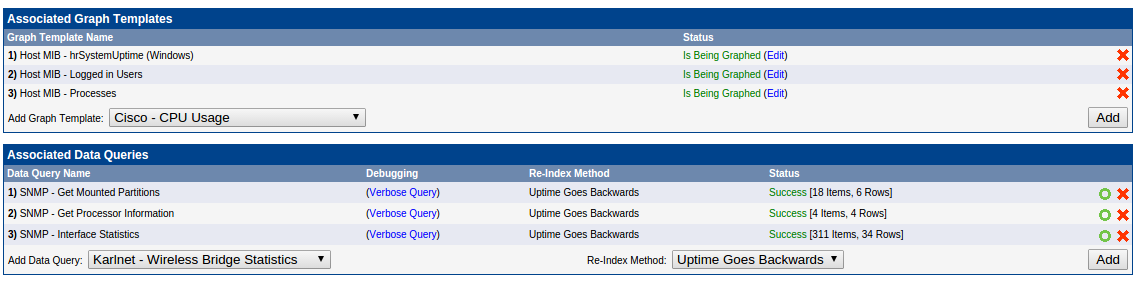
Console -> Devices -> Add để thêm các máy Client.

* **Windows Client:**

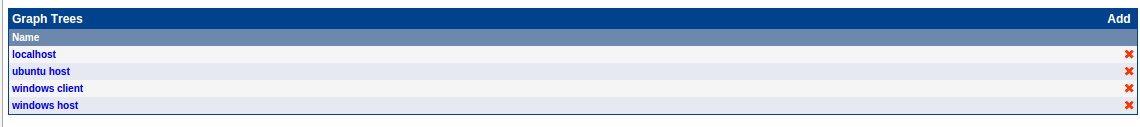
Điền các thông số:



->Add các thông số cần theo dõi:

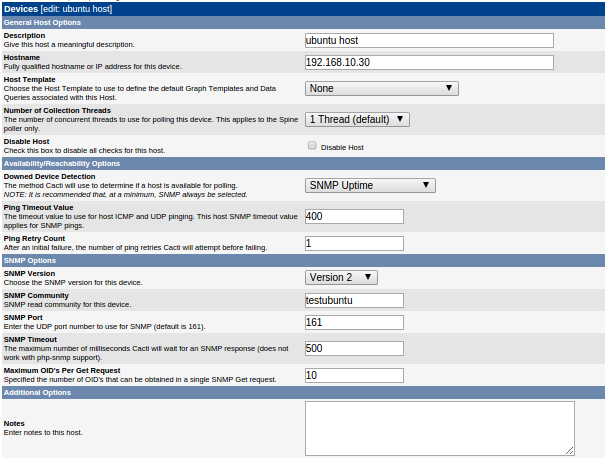


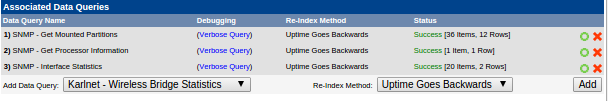
->Graph Trees để add windows host:

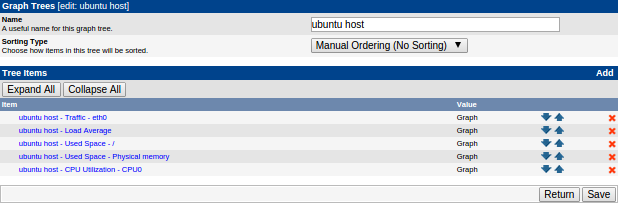


* **Ubuntu Client:**

Tương tự như Windows Client







**b.Cấu hình cacti server**

* **Cấu hình rsyslog**

Mặc định dữ liệu syslog bắt được của hệ thống sẽ chỉ lưu trữ ra file hệ thống. Để dữ liệu có thể lưu vào MySQL cần cấu hình lại rsyslog

sudo vim /etc/rsyslog.d/mysql.conf

Sửa nội dung file thành:

$ ModLoad ommysql

$template cacti\_syslog,"INSERT INTO syslog\_incoming(host, sourceip, facility, priority, sys\_date, message, status) values ('%hostname:::lower%', '%fromhost-ip%','%syslogfacility-text%', '%syslogpriority-text%', '%timereported:::date-mysql%', '%msg%', '0')", SQL

\*.\* >mysql\_host,mysql\_db,mysql\_user,mysql\_password;cacti\_syslog

Tạo bảng syslog\_incoming trong cơ sở dữ liệu:

CREATE TABLE `syslog\_incoming` (

`id` int(10) unsigned NOT NULL DEFAULT '0',

`host` varchar(128) DEFAULT NULL,

`sourceip` varchar(45) NOT NULL,

`facility` varchar(10) DEFAULT NULL,

`priority` varchar(10) DEFAULT NULL,

`sys\_date` datetime DEFAULT NULL,

`message` varchar(255) DEFAULT NULL,

`status` tinyint(4) NOT NULL DEFAULT '0',

`alert` tinyint(3) NOT NULL DEFAULT '0'

) ENGINE=InnoDB DEFAULT CHARSET=latin1

Khởi động lại dịch vụ rsyslog

* **Cấu hình snmp**

Gói phần mềm snmp bao gồm hai dịch vụ là snmpd và snmptrapd. Snmp lắng nghe trên cổng 161 và phản hồi các request snmp get, smpt GetRequest, SetRequest, GetNextRequest, GetBulkRequest, Response.

Snmptrapd phản hồi cho các request Trap. Mặc đình thì snmptrap daemon không được cấu hình để hoạt động, ta cần sửa lại cấu hình của snmpd

sudo vim /etc/default/snmpd

sửa lại dòng cấu hình:

TRAPDRUN=no

Thành:

TRAPDRUN=yes

Sửa lại file cấu hình:

sudo vim / etc/snmp/snmptrapd.conf

Nhập nội dung sau:

traphandle default /usr/sbin/snmptt # dữ liệu snmptrap được xử lý bằng snmptt

donotlogtraps no # lưu lại log snmptrap

tiếp tục sửa file /etc/snmp/snmptt.ini

sudo vim /etc/snmp/snmptt.ini

thông số **mysql\_dbi\_enable** ta set giá trị bằng **1**

Các thông số **mysql\_dbi\_host, mysql\_dbi\_port, mysql\_dbi\_database, mysql\_dbi\_table, mysql\_dbi\_table\_unknown, mysql\_dbi\_username, mysql\_dbi\_password** ta nhập vào các giá trị tương ưng

Set giá trị **log\_enable** bằng **1**

Khởi động lại dịch vụ snmpd, snmptt

# II.Thực nghiệm

....

# III.Kết quả thực nghiệm

....

# IV.Đánh giá

## 1.Ưu điểm

....

## 2.Nhược điểm

....

## 3.Hướng phát triển

....

**KẾT LUẬN**

............................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

# TÀI LIỆU THAM KHẢO

**[1]. Diep Thanh Nguyen, 04-2010, SNMP toàn tập.**

[2]. Nguyễn Văn Tam*, Bài giảng về mạng máy tính và kiến trúc SNMP*.

**[3].** <http://www.cacti.net/index.php>

[4]. Cacti 0.8 Learn Cacti and design a robust Network Operations Centre\_Thomas Urban\_3/2011

[5]. Cacti 0.8 Network Monitoring\_Dinangkur Kundu & S.M.Ibrahim Lavlu\_8/2009