TÌM HIỂU VÀ TRIỂN KHAI KAMAILIO

Nhóm L52

1. **Danh sách thành viên và công việc**

|  |  |  |  |
| --- | --- | --- | --- |
| Họ và tên | Mã SV | Công việc | Tiến độ |
| Nguyễn Thế Vinh | 175A071562 | - Cài đặt Kamailio  - Hướng dẫn cài đặt  - Hướng dẫn quản trị Kamailio  - Hướng dẫn phát triển Kamailio | - Cài đặt hoàn thành  - Hoàn thành  - 20% |
| Nguyễn Khắc Kỷ | 175A071312 | - Cài đặt Kamailio  - Tìm hiểu tài liệu về Kamailio : khái niệm, chức năng, hoạt động.  - Phân tích ưu, nhược điểm  - Hướng dẫn phát triển Kamailio | - Cài đặt hoàn thành  - Hoàn thành  - Hoàn thành |

1. **Nội dung nghiên cứu**
   1. **Kamailio là gì?**

* Kamailio (trước đây là OpenSER) là một máy chủ SIP được cấp phép theo giấy phép công cộng GNU.
* Nó có thể được thiết lập thành một nhà đăng ký dịch vụ SIP, hay là một máy chủ proxy hay một server chuyển hướng và và với những thứ đang được hỗ trợ như RADIUS / syslog để tính toán và ủy quyền, XML và JSON để điều hướng, SQL backend, …
* Kamailio là một từ Hawai có nghĩa là “giao tiếp”. Kamailio được chọn vì những đặc tính riêng biệt của riêng nó.
* Như vậy, SIP là gì ?
* SIP (Session Initiation Protocol - giao thức khởi tạo phiên) là một giao thức truyền thông để truyền tín hiệu và điều khiển các phiên truyền thông trong các ứng dụng điện thoại sử dụng internet cho các cuộc gọi có tiếng hoặc cả video trong hệ thống điện thoại, cũng như nhắn tin qua mạng Internet Protocol (IP).
* Giao thức này định dạng cụ thể các tin nhắn trao đổi với sự hợp tác của những người tham gia đa phương tiện.
* SIP được thiết kế độc lập với các giao thức ở tầng giao vận (Transport Layer ) nằm ở dưới và có thể được sử dụng chung với UDP, TCP hay SCTP.
* SIP thường dùng chung với UDP vì cùng chung mục đích truyền video cần tính nhanh chóng không nhất thiết phải bảo toàn dữ liệu.

Tóm lại : SIP là một giao thức khởi tạo phiên nó được dùng để truyền và điều khiển các đa phương tiện (giọng nói, video, ...) bằng cách thức nào ? : qua mạng IP.

* 1. **Kamailo có chức năng cụ thể là gì ?**
* Kamailio có thể được thiết lập là một nhà đăng ký dịch vụ SIP, cũng có thể là một máy chủ proxy vậy chúng ta xem nó hoạt động như thế nào, hay chức năng của nó ra sao?

Vì Kamailio là giao thức sử dụng SIP → ta cần biết cơ bản về SIP proxy

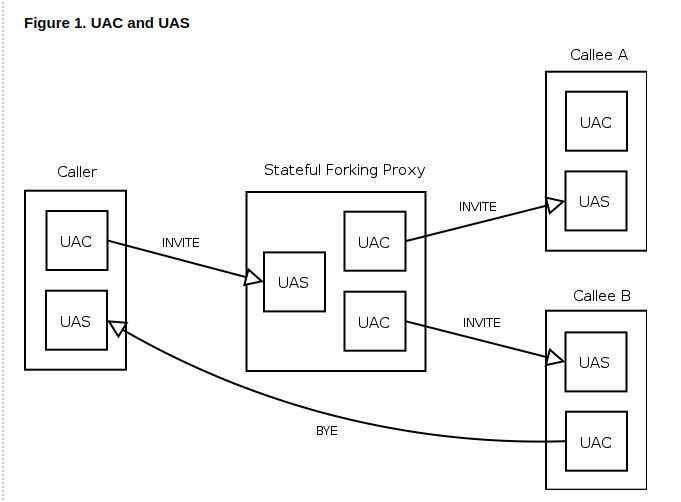
* Proxy là gì?

Giả sử ta có 2 người dùng A gọi cho B, lời gọi sẽ không được chuyển trực tiếp đến B mà ta sẽ chuyển qua trung gian 1 máy chủ proxy để điều hướng các phiên và cũng như thuận lợi cho các vấn để xác thực người dùng.

Ta hãy xem với một ví dụ về cấu trúc mạng proxy Server

1. User Agent : thường được biết đến như User Agent Server (UAS) và User Agent Client (UAC)

* Với mỗi người dùng (UserAgent) đều có 1 UAC và UAS
* UserAgent hành động như là một UAC khi gửi yêu cầu (request) và nhận phản hồi (response)
* UserAgent thay đổi như là UAS khi người được gọi gửi lại tín hiệu BYE và hủy phiên



1. Proxy Servers (đây là thứ chúng ta quan tâm)

SIP cho phép chúng ta tạo một hệ thống mạng lưới hosts gọi là proxy servers. UserAgents có thể gửi tín hiệu tới proxy servers .Proxy servers góp phần quan trọng trong hệ thống SIP . Nó có trách nhiện điều hướng một lời mời gọi phiên (session invitation) thẳng tới người được nhận và người nhận sẽ quyết định xem chấp nhận hay từ chối cuộc gọi.

Có 2 loại SIP proxy servers cơ bản:

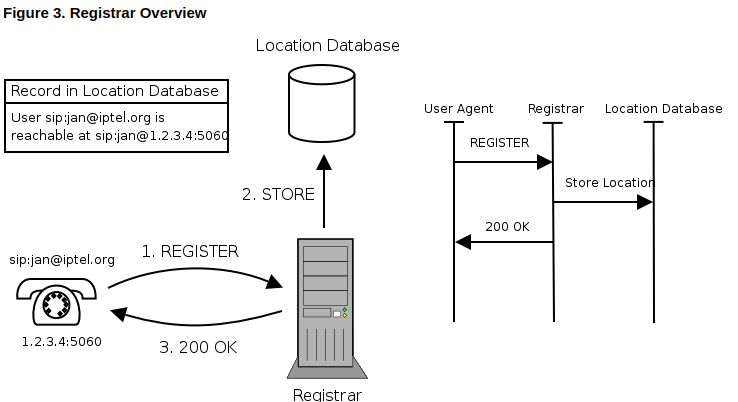
* Stateless Servers: đối với mô hình này proxy chỉ làm nhiệm vụ chuyển tiếp đơn giản. Nó chỉ thực hiện để chuyển tiếp mỗi request và loại bỏ hoàn toàn các thông tin về request mỗi khi đã chuyển đi.
* Stateful Servers : Stateful proxy thì phức tạp hơn. Mỗi lần server nhận được yêu cầu, stateful proxy khởi tao một trạng thái và giữ trạng thái đó cho đến khi chuyển giao kết thúc .Một vài chuyển giao đặc biệt là INVITE, có thể kết thúc muộn (chờ đến khi người nhận chấp thuận hay từ chối). Bởi vì thế stateful proxy phải duy trì trạng thái trong lúc chuyển giao cho dù có 1 vài giới hạn.

Một tính năng đặc biệt của stateful proxy là có thể lựa chọn “fork” 1 request có nghĩa là định tuyến request đến nhiều đích khác nhau.

1. SIP Registrar Server

Chúng ta đã đề cập tới cách server SIP điều hướng nhưng chúng ta chưa biết làm sao proxy có thể biết được địa chỉ hiện tại của người dùng.

Như trong hình Figure 3 một SIP phone muốn đăng ký với nhà đăng kí (registrar). Nhà đăng kí là một thực thể đặc biệt của SIP nhậ những đơn đăng ký của người dùng (user), giải nén những thông tin về địa chỉ hiện tại của họ (như là địa chỉ IP ,cổng đang dùng và tên người dùng) và lưu chúng trên cơ sở dữ liệu.



* Figure 3 thể hiện một đăng ký SIP thông thường. Một tín hiệu đăng ký chứa địa chỉ là sip:jan@iptel.org và có địa chỉ là 1.2.3.4:5060 được gửi đến nhà đăng kí. Nhà đăng kí này giải nén tất cả những thông tin ấy và lưu nó trong cơ sở dữ liệu . Nếu tất cả ổn thỏa thì nhà đăng kí sẽ gửi tín hiểu 200:OK về điện thoại và tiến trình kết thúc.

Mỗi lần đăng ký đều có thời gian sống nhất định. Trường Expires Header sẽ quyết định lần đăng ký này sống bao lâu. User Agent phải làm mới đăng ký của họ nếu không sẽ hết hạn và người dùng sẽ không còn khả dụng.

* Những tính năng khác mà Kamailio hỗ trợ:
* Hỗ trợ giao tiếp thông qua UDP, TCP , TLS và SCTP.
* Ipv4 và Ipv6.
* Xử lý tiến trình không đồng bộ TCP.
* Bảo mật giao tiếp IP và Network Authentication…

Tóm lại : Kamailio sử dụng giao thức SIP để :

* Điều hướng
* Tạo phiên
* Đăng kí SIP
* Truyền và điều hướng đa phương tiện (video, giọng nói, cuộc gọi…)
  1. **Hoạt động của Kamailio hoặc So sánh Ưu/Nhược điểm**
* Ưu/Nhược điểm :
* Kamailio Server dựa trên SIP (giao thức khởi tạo phiên): giao thức này là một giao thức gửi tín hiệu để khởi tạo duy trì, ngắt phiên trong thế giới thực những thứ như :giọng nói ,video, và các ứng dụng tin nhắn.
* SIP có thể được sử dụng để khởi tạo phiên cho nhóm-2(unicast) hay nhóm-nhiều (multicast).
* Cho phép tùy chỉnh những cuộc gọi đang tồn tại. Sự điều chỉnh này có thể liên quan đến địa chỉ hoặc cổng nguồn, mời nhiều người tham dự.
* Kamailio bạn có một cộng đồng người dùng hoạt động rộng rãi, phát triển nhanh chóng, những nhà phát triển thân thiện và sẵn sàng giúp
* So sánh với một vài server SIP khác như Opensips thì Kamailio có kho modules nhiều hơn hẳn.

**d. Hướng dẫn cài đặt Kamailio trên Ubuntu**

Bước 1 : Cài đặt máy chủ Apache2 HTTP

* Cài đặt Apache2 HTTP

|  |
| --- |
| #sudo apt update    #sudo apt install apache2 |

* Sau khi cài đặt Apache2 thì khởi động 3 lệnh dừng , khởi động và cho phép dịch vụ Apache2 luôn khởi động với máy chủ khởi động.

|  |
| --- |
| #sudo systemctl stop apache2.service  #sudo systemctl start apache2.service  #sudo systemctl enable apache2.service |

* Để kiểm tra thiết lập Apache2, hãy mở trình duyệt của bạn và duyệt đến tên máy chủ hoặc địa chỉ IP của máy chủ

http: //localhost

|  |
| --- |
|  |

Bước 2 : Cài đặt máy chủ cơ sở dữ liệu MariaDB

* Cài đặt MariaDB Database Server

|  |
| --- |
| #sudo apt install mariadb-server mariadb-client |

* Sau khi cài đặt MariaDB thì khởi động 3 lệnh dừng , khởi động và cho phép dịch vụ MariaDB luôn khởi động khi máy chủ khởi động

|  |
| --- |
| #sudo systemctl stop mariadb.service  #sudo systemctl start mariadb.service  #sudo systemctl enable mariadb.service |

* Sau đó, chạy các lệnh bên dưới để bảo mật máy chủ MariaDB bằng cách tạo mật khẩu gốc và không cho phép truy cập root từ xa

|  |
| --- |
| #sudo mysql\_secure\_installation |

Khi được nhắc, hãy trả lời các câu hỏi dưới đây bằng cách làm theo :

* + Enter current password for root (enter for none): Just press the Enter
  + Set root password? [Y/n]: Y
  + New password: Enter password
  + Re-enter new password: Repeat password
  + Remove anonymous users? [Y/n]: Y
  + Disallow root login remotely? [Y/n]: Y
  + Remove test database and access to it? [Y/n]:  Y
  + Reload privilege tables now? [Y/n]:  Y

|  |
| --- |
|  |

* Để kiểm tra xem MariaDB đã được cài đặt chưa, hãy nhập các lệnh bên dưới để đăng nhập vào máy chủ MariaDB

|  |
| --- |
| #sudo mysql -u root -p |

Bước 3 : Cài đặt PHP các Modules liên quan

* Cập nhật

|  |
| --- |
| #sudo apt update |

* Cài đặt PHP 7.2 và các modules liên quan

|  |
| --- |
| #sudo apt install php7.2 libapache2-mod-php7.2 php7.2-common php7.2-curl php7.2-intl php7.2-mbstring php7.2-xmlrpc php7.2-mysql php7.2-gd php7.2-imagick php-pear php7.2-xml  Php7.2-cli php7.2-zip php7.2-sqlite |

* Mở tệp cấu hình mặc định PHP cho Apache2

|  |
| --- |
| #sudo nano /etc/php/7.2/apache2/php.ini |

* Thay đổi các giá trị như bảng rồi lưu

|  |
| --- |
| * file\_uploads = On      * allow\_url\_fopen = On      * short\_open\_tag = On      * memory\_limit = 256M      * upload\_max\_filesize = 100M      * max\_execution\_time = 360      * max\_input\_vars = 1500      * date.timezone = Asia.HoChiMinh |

* Khởi động lại Apache2

|  |
| --- |
| #sudo systemctl restart apache2.service |

* Kiểm tra cài đặt PHP 7.2 với Apache2, hãy tạo tệp phpinfo.php trong thư mục gốc Apache2

|  |
| --- |
| #sudo nano /var/www/html/phpinfo.php |

* Nhập nội dung vào tệp và lưu

|  |
| --- |
| <php phpinfo( ) ; ?> |

* Sau đó duyệt đến máy chủ của bạn : <http://localhost/phpinfo.php>

|  |
| --- |
|  |

Bước 4 : Tải Kamailio

* Tải xuống và thêm khóa GPG cho kho lưu trữ của nó

|  |
| --- |
| #wget -0- <http://deb.kamailio.org/kamailiodebkey.gpg> | sudo apt-key add – |

* Tạo tệp lưu trữ

|  |
| --- |
| #sudo nano /etc/apt/sources.list.d/kamailio.list |

* Chép, dán và lưu vào tệp

|  |
| --- |
| #deb http://deb.kamailio.org/kamailio52 bionic main  #deb-src http://deb.kamailio.org/kamailio52 bionic main |

* Cập nhật các gói Ubuntu và cài đặt Kamailio

|  |
| --- |
| #sudo apt update    #sudo apt install kamailio kamailio-mysql-modules kamailio-websocket-modules |

* Mở tệp cấu hình Kamailio

|  |
| --- |
| #sudo nano /etc/kamailio/kamctlrc |

* Chỉnh sửa trong tệp và lưu

|  |
| --- |
| SIP\_DOMAIN = kamailio.example.com  DBENGINE = MYSQL |

* Tạo cơ sở dữ liệu, người dùng và các bảng cần thiết cho Kamailio

|  |
| --- |
| #kamdbctl create |

* Nếu quyền truy cập bị từ chối bởi **root@localhost**

|  |
| --- |
|  |

thì :

|  |
| --- |
| * Đăng nhập vào máy chủ MariaDB chạy lệnh :   #sudo mysql -u root   * Tắt xác thực plugin cho người dùng root   use mysql;  update user set plugin='' where User='root';  flush privileges;  exit;   * Khởi động lại   #sudo systemctl restart mariadb.service |

* Bây giờ hãy chạy tập lệnh Kamailio để tạo cơ sở dữ liệu và người dùng

và trả lời các cài đặt bên dưới

|  |
| --- |
| * Enter character set name:   latin1   * INFO: creating database kamailio ... * INFO: granting privileges to database kamailio ... * INFO: creating standard tables into kamailio ... * INFO: Core Kamailio tables succesfully created. * Install presence related tables? (y/n): y * INFO: creating presence tables into kamailio ... * INFO: Presence tables succesfully created. * Install tables for imc cpl siptrace domainpolicy carrierroute * drouting userblacklist htable purple uac pipelimit mtree sca mohqueue rtpproxy rtpengine? (y/n): y * INFO: creating extra tables into kamailio ... * INFO: Extra tables succesfully created. * Install tables for uid\_auth\_db uid\_avp\_db uid\_domain   uid\_gflags uid\_uri\_db? (y/n): y   * INFO: creating uid tables into kamailio ... * INFO: UID tables succesfully created. |

* Mở /etc/kamailio/kamailio.cfg bằng cách chạy lệnh

|  |
| --- |
| #sudo nano /etc/kamailio/kamailio.cfg |

* Thêm và lưu vào tệp sau #! KAMAILIO

|  |
| --- |
| #!define WITH\_MYSQL  #!define WITH\_AUTH  #!define WITH\_USRLOCDB  #!define WITH\_ACCDB |

* Áp dụng các lệnh thay đổi

|  |
| --- |
| #sudo systemctl restart kamailio |

Bước 5 : Cài đặt và cấu hình Siremis

* Cài đặt XML\_RPC

|  |
| --- |
| #sudo pear install XML\_RPC2 |

* Thay đổi thư mục thành root web Apache, cài đặt git và tải xuống các gói Siremis.

|  |
| --- |
| #cd / var / www /  #sudo apt install git    #sudo git clone <https://github.com/asipto/siremis> |

* Vào thư mục Siremis và chuẩn bị môi trường

|  |
| --- |
| #cd /var/www/siremis  #sudo make prepare24  #sudo make chown |

* Định cấu hình tệp cấu hình trang Apache2 cho Siremis

|  |
| --- |
| #sudo nano /etc/apache2/sites-available/siremis.conf |

* Chèn vào tập tin và lưu. Thay thế tên miền (ServerName) và vị trí thư mục gốc (ServerAlias)

|  |
| --- |
| <VirtualHost \*:80>  ServerAdmin admin@example.com  DocumentRoot /var/www/siremis/siremis  Alias /siremis "/var/www/siremis/siremis"  ServerName kamailio.example.com  ServerAlias www.kamailio.example.com  <Directory "/var/www/siremis/siremis">  Options Indexes FollowSymLinks MultiViews  AllowOverride All  Require all granted    <FilesMatch "\.xml$">  Require all denied  </FilesMatch>  <FilesMatch "\.inc$">  Require all denied  </FilesMatch>  </Directory>  ErrorLog ${APACHE\_LOG\_DIR}/error.log  CustomLog ${APACHE\_LOG\_DIR}/access.log combined  </VirtualHost> |

Bước 6 : Kích hoạt Modules và viết

* Sau khi định cấu hình Virtualhost ở trên, bật nó :

|  |
| --- |
| #sudo a2ensite siremis.conf  #sudo a2enmod rewrite  #sudo systemctl restart apache2.service |

* Cấp quyền truy cập cho người dùng Siremis vào cơ sở dữ liệu của nó

|  |
| --- |
| #sudo mysql -u root -p |

* Chạy lệnh SQL

|  |
| --- |
| * GRANT ALL PRIVILEGES ON siremis.\* TO siremis@localhost IDENTIFIED BY 'siremisrw'; * FLUSH PRIVILEGES; * EXIT; |

* Sau đó mở trình duyệt và duyệt đến tên miền máy chủ hoặc tên máy chủ

<http://kamailio.example.com/>

|  |
| --- |
|  |

* Xác thực rằng tất cả các yêu cầu được đáp ứng, sau đó tiếp tục với trình hướng dẫn.

|  |
| --- |
|  |

* Nhập thông tin cấu hình cơ sở dữ liệu

|  |
| --- |
|  |

* Thiết lập hoàn tất và cung cấp cho bạn tên người dùng và mật khẩu quản trị viên. Sử dụng thông tin đăng nhập và bắt đầu thiết lập môi trường của bạn

|  |
| --- |
|  |

* Tên đăng nhập : admin

Mật khẩu : admin

|  |
| --- |
|  |

* Hoàn thành thiết lập

|  |
| --- |
|  |

* 1. **Hướng dẫn quản trị Kamailio**

Thiết lập Module Call\_Cotrol của Kamailio để nói chuyện giữa hai người.

* Module này cho phép một người giới hạn thời lượng cuộc gọi và tự động kết thúc chúng khi vượt quá giới hạn áp đặt. Trường hợp sử dụng chính của nó là để thực hiện một hệ thống trả trước, nhưng nó cũng có thể được sử dụng để áp đặt giới hạn toàn cầu cho tất cả các cuộc gọi được xử lý bởi proxy.
* Một ứng dụng bên ngoài được gọi là call\_control theo dõi các cuộc gọi có giới hạn thời gian và tự động kết thúc chúng khi chúng vượt quá nó.

Ứng dụng này nhận được yêu cầu từ Kamailio và đưa ra yêu cầu cho công cụ xếp hạng để tìm hiểu xem cuộc gọi có cần được giới hạn hay không. Khi một cuộc gọi kết thúc (hoặc kết thúc), nó cũng sẽ hướng dẫn công cụ xếp hạng ghi nợ số dư cho người gọi với số tiền đã tiêu thụ.

Ứng dụng callcontrol có sẵn từ <http://callcontrol.ag-projects.com/>

* Công cụ xếp hạng được sử dụng để tính thời hạn dựa trên tín dụng của người gọi và giá đích và để ghi nợ số dư của người gọi sau khi cuộc gọi kết thúc.

Điều này có sẵn như là một phần của CDRTool từ <http://cdrtool.ag-projects.com/>

* Ứng dụng callcontrol chạy trên cùng một máy với Kamailio và chúng giao tiếp qua ổ cắm hệ thống tập tin, trong khi công cụ đánh giá có thể chạy trên một máy chủ khác và giao tiếp với ứng dụng callcontrol bằng kết nối TCP.
* Callcontrol được gọi bằng cách gọi hàm call\_control () cho HÓA ĐƠN ban đầu của mỗi cuộc gọi mà chúng tôi muốn áp dụng giới hạn. Điều này sẽ kết thúc như một yêu cầu cho ứng dụng điều khiển cuộc gọi, ứng dụng này sẽ kết hợp công cụ đánh giá trong một giới hạn thời gian cho người gọi và đích đã cho. Công cụ xếp hạng sẽ xác định xem điểm đến có bất kỳ chi phí liên quan nào không và liệu người gọi có bất kỳ giới hạn tín dụng nào không và nếu có sẽ trả lại thời gian anh ta được phép gọi điểm đến đó. Nếu không, nó sẽ chỉ ra rằng không có giới hạn liên quan đến cuộc gọi. Nếu có giới hạn, ứng dụng điều khiển cuộc gọi sẽ giữ lại phiên và gắn bộ hẹn giờ với nó sẽ hết hạn sau thời gian nhất định khiến nó gọi lại cho Kamailio với yêu cầu kết thúc hộp thoại. Nếu công cụ đánh giá trả về rằng không có giới hạn cho cuộc gọi, phiên sẽ bị loại bỏ bởi ứng dụng điều khiển cuộc gọi và nó sẽ cho phép nó tiến hành bất kỳ giới hạn nào. Một phản hồi thích hợp được trả về mô-đun call\_control sau đó được trả về bởi lệnh gọi hàm call\_control () và cho phép tập lệnh đưa ra quyết định dựa trên câu trả lời.