**STEP 4: MiddleWare cơ bản**

Phần này được chia làm 3 phần là

+ DB cơ sở

+ Web server

+ KVS cơ sở: <http://blog.innobase.co.jp/entry/2014/10/31/115327>

Nhiệm vụ cần làm: Tìm hiểu Middleware là gì ? Tác dụng của nó là gì?

**Phần 1: MiddleWare**

MiddleWare là gì?

Định nghĩa 1: <https://ja.wikipedia.org/wiki/%E3%83%9F%E3%83%89%E3%83%AB%E3%82%A6%E3%82%A7%E3%82%A2>

MiddleWare là phần mềm được cho vào nằm giữa phần mềm ứng dụng tổ chức quản lý các nghiệp vụ và Hệ điều hành tổ chức kiểm soát các chức năng cơ bản của máy tính , trong phân dã(???) của computer.

Thông thường,nó bao gồm các thành phần có chức năng thu thập những chức năng mang tính cộng đồng của phần mềm ứng dụng hoặc là khuyech đại khả năng của hệ điều hành. Khi đưa ra yêu cầu đối với ứng dụng(application software) thì middleware sẽ đưa ra yêu cầu quan trọng đối với hệ điều hành, kết quả sẽ được gửi phản hồi tới ứng dụng . Hoặc là trong bản thân middleware đã tiến hành tổ chức kiểm soát những nghiệp vụ như là khởi động , cấm đoán, giám sát của các ứng dụng.

Đại biểu của middleware là hệ thống quản lý database (DBMS) và transaction monitor

Định nghĩa 2:<http://e-words.jp/w/%E3%83%9F%E3%83%89%E3%83%AB%E3%82%A6%E3%82%A7%E3%82%A2.html>

Middleware là thứ cung cấp chức năng được sử dụng để chia sẻ từ rất nhiều software, nó được đặt ở giữa application software và operating system trong từng loại software.

Định nghĩa từ Wikipedia:

Middleware là phần mềm cung cấp một mối liên kết giữa các phần mềm ứng dụng khác nhau. Phần mềm loại này bao gồm một tập các dịch vụ cho phép sự tương tác giữa các tiến trình chạy trên một hoặc nhiều máy khác nhau. Middleware đôi khi còn được gọi là đường ống (plumping) vì nó kết nối hai ứng dụng với nhau và truyền dữ liệu giữa chúng. Middleware cho phép một cơ sở dữ liệu truy nhập dữ liệu tại một cơ sở dữ liệu khác.

Định nghĩa từ ObjectWeb :

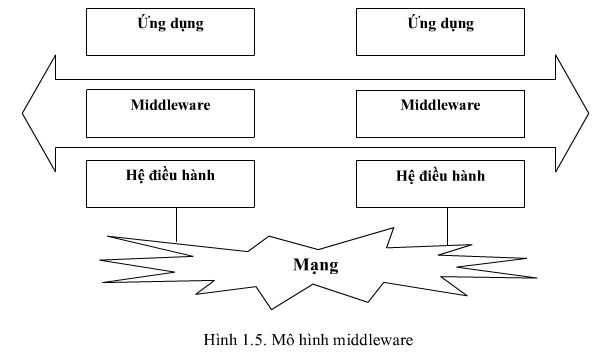
Tầng phần mềm nằm giữa hệ điều hành và ứng dụng ở mỗi đầu của một hệ thống phân tán nằm trong mạng máy tính

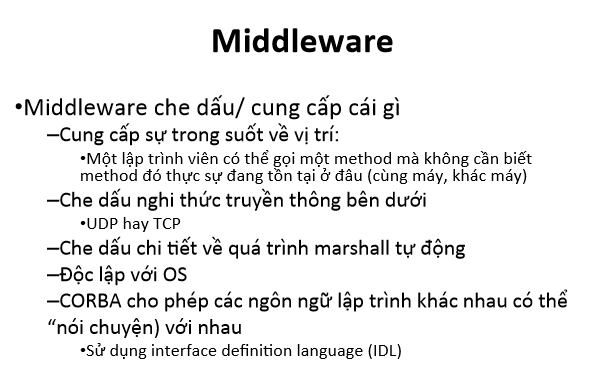
Kết luận:

Middleware chính là lớp phần mềm trung gian nằm giữa tầng mạng và tầng ứng dụng nhằm cung cấp các dịch vụ, giúp các thành phần có thể tương tác với nhau mà ko phụ thuộc vào thành phần xây dựng theo nền OS gì.

Mô hình phầm mềm trung gian MiddleWare

Vì sao phải có middleware ? Cùng với sự phát triển về internet thì hệ thống thông tin cũng phát triển và nó đóng vai trò quan trọng giúp mọi người có thể giao tiếp, muc bán, kinh doanh.Nó mang đến cả sự đa dạng về ọi mặt, giảm thiểu những khó khan như khoảng cách hay bất đồng ngôn ngữ. Tuy nhiên khi xây dựng những ứng dụng đáp ứng điều này thì các nhà phát triển đã gặp phải những khó khan về việc ko tương thích phần cứng hay phần mềm do các hệ điều hành khác nhau cung cấp.Nếu đi xây dựng các ứng dụng tương thích với từng hệ điều hành thì cần rất nhiều tiền của cũng như đôi khi ko đạt được kết quả mong muốn. Và middleware ra đời ! Bằng việc cung cấp tập các chuẩn cho việc xử lý phân tán, middleware đã loại bỏ những khó khan về sự k tương thích phần cứng cũng như phần mềm hay hệ điều hành gây ra.Như vậy ta có thể tạm hiểu, middleware là 1 tầng trung gian đặt giữa tầng ứng dụng và tầng mạng, giúp cho các ứng dụng giao tiếp với nhau mà ko cần chú ý đến các tầng khác thấp hơn.





**Ngày 08/09/2015**

Tài liệu 1: Ví dụ cụ thể về Middleware

+ <http://oshiete.goo.ne.jp/qa/2850501.html>

Tại mức sơ cấp System Admin, thành phần trung gian giữa Application Software và OS được gọi là Middleware như là

+ Hệ thống quản lý database(DBMS)

+ Hệ thống quản lý truyền tin(Bao gồm kiểm soát LAN)

+ Tool hỗ trợ phát triển software

+ Tool EUC

+ Tool quản lý vận dụng

Sẽ được giới thiệu trong bài viết này.Có những thứ cố gắng có thể suy đoán ra ví dụ cụ thể nhưng cũng có những thứ ko thể liên tưởng đến hình ảnh nào cả.Vì vậy ở bài này các bạn hãy giúp tôi xác nhận xem suy đoán của mình là đúng hay sai, với những trường hợp ko thể liên tưởng được thì hãy giúp tôi có những dẫn chứng cụ thể.

+ Hệ thống quản lý database(DBMS- Database Manager System)

Có lẽ đó là những thứ như là Oracle , MySQL

+ Hệ thống quản lý truyền tin(bao gồm kiểm soát LAN)

Các application tiến hành kiểm soát truyền tin như là TCP/IP driver, firmware cũng liên quan. Chứ không phải các loại application do người dùng thông thường thao tác

+Tool hỗ trợ phát triển software:

Đó là các tool hỗ trợ việc lập trình bằng cách ngôn ngữ lập trình như C ,C++ , Perl,..

+ Tool EUC

Application được người dùng cuối sử dụng(end user) như là Word , Excel

+ Tool quản lý vận dụng

Là các tool quản lý PC client, tool quản lý DB system , và quản lý server của mail,web hay là tool quản ký cơ khí vận dụng(???) Gần đây thì các tool ngăn chặn việc rò rỉ thông tin cũng xuất hiện nhiều.

Middleware là phần mềm tổ chức các hành động, quản lý ở giữa operating và application software giữa các loại software.

Middle là 1 lớp nằm giữa hệ điều hành và các ứng dụng phân tán. Che giấu sự phức tạp và ko đồng nhất của hệ thống phân phối.

**Database**

**1.Dữ liệu và thông tin**

Thông tin: là sự phản ảnh các hiện tượng sự vật của thế giới khách quan bằng nhiều hình thức khác nhau như hình ảnh, âm thanh, ký tự,..

Dữ liệu: là thông tin được lưu trữ trong các thiết bị lưu trữ.Vì vậy ta có thể truy nhập vào dữ liệu và trích xuất ra thông tin.Hàng ngày có vô vàn thông tin được sinh ra, nhưng ko phải thông tin nào cũng được coi là dữ liệu.Chỉ những thông tin được lọc ra và lưu trữ tùy theo mục đích sử dụng mới được gọi là dữ liệu. Vì vậy mà có thể đối với tổ chức, cá nhân này thì thông tin ko phải là dữ liệu(không cần chọn lọc và lưu trữ), nhưng đôi với tổ chức, cá nhân khác thì nó chính là dữ liệu(đã được chọn lọc và lưu trữ). Dữ liệu tồn tại dưới nhiều hình thức khác nhau như là hình ảnh, âm thanh, ký hiệu,hay các con số, chữ cái…

Bản thân dữ liệu không có ý nghĩa với con người, nhưng thông tin mà nó lưu trữ lại có ý nghĩa với con người.

**2. File Base System(Hệ thống quản lý bằng tập tin)**

Có nhiều cách lưu trữ thông tin như là viết ra, ghi âm,..Nhưng thông thường để lưu trữ thông tin trên máy tính, ta sẽ lưu chúng dưới dạng các file hay còn gọi là các tập tin. Với phương pháp này thì sẽ có những ưu điểm là tốc độ triển khai nhanh, trực quan với những người ko có chuyên môn về công nghệ. Tuy nhiên nó cũng có những nhược điểm nổi bật như là: dữ liệu không nhất quán, đôi khi có nhiều người cùng sử dụng 1 file sẽ gây ra hiện tượng lưu đè , trùng lặp dữ liệu nhiều, tính chia sẻ dữ liệu không cao(có thể nói là kém)

Hiện nay đa số người dùng máy tính đều sử dụng cách lưu trữ thông tin này cho riêng mình, nên nó đã trở nên phổ biến. Nhưng nếu với các hệ thống cần nhiều dữ liệu và cần tìm kiếm,cần truy xuất dữ liệu tức thời hay bảo trì dữ liệu thì cách này có hiệu quả hay không ? Rõ ràng là không. Chính vì thế mà khái niệm cơ sở dữ liệu và hệ quản trị cơ sở dữ liệu ra đời.

🡪 Qua phân tích trên, chúng ta nhận thấy việc tổ chức dữ liệu theo hệ thống tệp tin hoàn toàn không phù hợp với những hệ thống thông tin lớn. Việc xây dựng một hệ thống thông tin đảm bảo được tính nhất quán dữ liệu, đáp ứng được nhu cầu khai thác đồng thời của nhiều người là thực sự cần thiết.

**3.Tổng quan về hệ cơ sở dữ liệu**

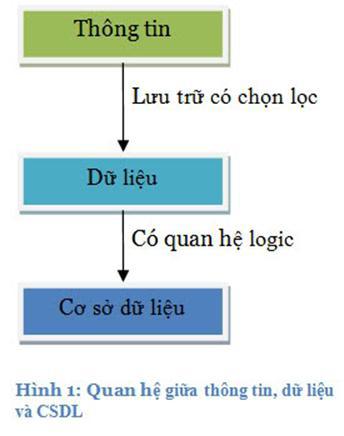
**3.1.Cơ sở dữ liệu(Database)**

CSDL là tập hợp các “dữ liệu có cấu trúc” và liên quan với nhau được lưu trữ trên máy tính, được nhiều người sử dụng và được “tổ chức theo một mô hình”.

P/s: Thế nào là dữ liệu có cấu trúc? (???)

Điểm nhấn mạnh ở đây đó là CSDL là tập hợp các thông tin có tính chất hệ thống chứ không phải các thông tin rời rạc, ko liên quan đến nhau.Các thông tin này phải có cấu trúc và tập hợp các thông tin này phải đáp ứng nhu cầu khai thác của nhiều người một cách đồng thời.

Nó được xây dựng để phục vụ mục đích lưu trữ, truy xuất dữ liệu, hay cập nhật dữ liệu cho ứng dụng nào đó hay chính bản thân người dùng với các mục đích khác nhau.





**3.2.Ưu , nhược điểm của CSDL**

Qua định nghĩa về CSDL, ta nhận thấy ngay các ưu điểm của nó

Về bản thân thông tin lưu trữ:

+ Giảm sự trung lặp thông tin ở mức thấp nhất, do đó đảm bảo tính nhất quán và toàn vẹn dữ liệu

+ Đảm bảo dữ liệu có thể được truy xuất theo nhiều cách khác nhau

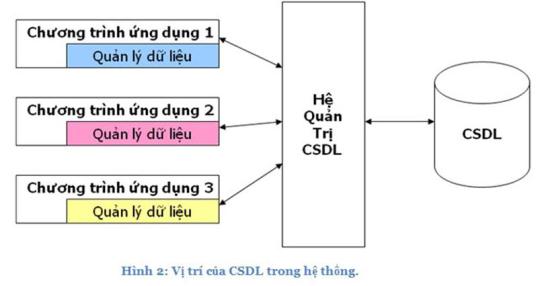
+ Khả năng chia sẻ thông tin cho nhiều người sử dụng và nhiều ứng dụng khác nhau

Về hiệu quả sử dụng thông tin:

+ Chia se thông tin cho nhiều người dùng khác nhau

+ Tiết kiệm tài nguyên

+ Tăng hiệu quả khai thác



Tuy nhiên để đảm bảo điều đó thì cũng có những vấn đề nảy sinh sau đây:

Cần xác định rõ trách nhiệm dối với sự an toàn của dữ liệu, tính chính xác của dữ liệu(ai có trách nhiệm cập nhật, chỉnh sửa dữ liệu và những thông tin nào được chia sẻ)

Cần 1 cơ chế bảo mật hay phân quyền khai thác thông tin

Giải quyết sự tranh chấp trong truy cập dữ liệu khi có nhiều người dùng cùng truy cập đến 1 nguồn dữ liệu.

CSDL cũng có chu kỳ sống tương tự như phần mềm nên việc phân tích, thiết kế cần đòi hỏi người có chuyên môn và nhiều kinh nghiệm

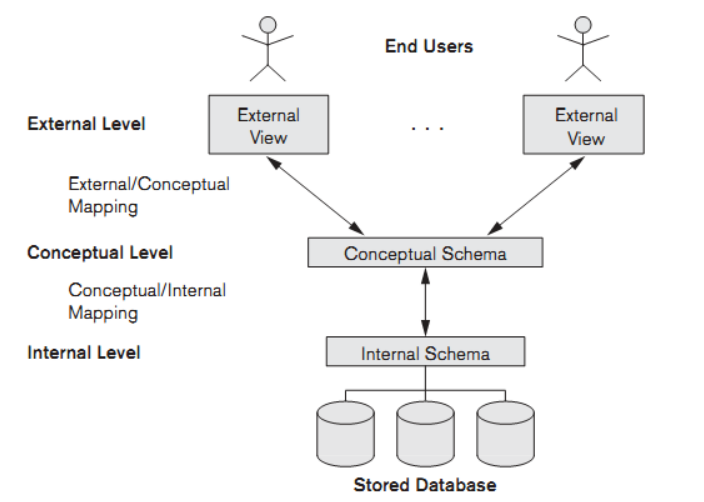
**3.3. Ba mức biểu diễn của một CSDL**

Để đảm bảo tính hiệu quả của hệ thống, đòi hỏi phải thiết kế các CSDL phức tạp để biểu diễn dữ liệu. Và để che dấu sự phức tạp này thì các nhà phát triển đã thông qua các mức trừu tượng để đơn giản hóa sự trao đổi của người sử dụng với hệ thống. Có 3 mức:

+ Mức vật lý: Mức thấp nhất của sự trừu tượng, mô tả dữ liệu hiện được lưu trữ thế nào. Ở mức này, cấu trúc dữ liệu ở mức thấp, phức tạp và được mô tả chi tiết.

+ Mức logic: Dữ liệu nào được lưu trữ trong CSDL và mối quan hệ giữa chúng. Mức logic được sử dụng bởi những người quản trị CSDL.

+ Mức khung nhìn: Các khung nhìn khác nhau của những người sử dụng đặt vào CSDL. View sẽ làm đơn giản hóa sự trao đổi của người dùng với hệ thống.



3.4. Các đối tượng sử dụng CSDL

+ Người sử dụng ko chuyên về lĩnh vực tin học và CSDL: CSDL cần có những công cụ để những đối tượng này có thể sử dụng và khai thác khi cần.

+ Chuyên viên tin học biết khai thác CSDL: Những người này có thể xây dựng những ứng dụng khác nhau phục vụ cho nhiều mục đích khác nhau trên CSDL

+ Người quản trị CSDL: Là người hiểu biết về tin học và các hệ quản trị CSDL và hệ thống máy tính. LÀ người tổ chức CSDL(khai báo cấu trúc, ghi nhận yêu cầu bảo mật) , là người cấp quyền khai thác CSDL.

**4. Hệ quản trị CSDL(DBMS)**

**4.1.Hệ quản trị cơ sở dữ liệu là gì**

**Hệ quản trị dữ liệu** là phần mềm:

+ Cung cấp môi trường tạo lập cơ sở dữ liệu(Định nghĩa các cấu trúc để lưu trữ thông tin trên máy)

+ Cung cấp môi trường cập nhật và khai thác dữ liệu(Nhập dữ liệu, thao tác trên các dữ liệu đảm bảo sự an toàn và bí mật của dữ liệu)

+ Cung cấp công cụ kiểm soát, điều khiển truy cập vào CSDL

Định nghĩa cấu trúc CSDL bao gồm việc xác định kiểu dữ liệu, cấu trúc, và những ràng buộc cho dữ liệu được lưu trữ trong CSDL

Nhập dữ liệu:Là việc lưu trữ dữ liệu vào các thiết bị lưu trữ trung giạn được điều khiển bằng hệ quản trị CSDL

Thao tác dữ liệu: bao gồm những chưc năng truy xuất CSDL để tìm kiếm thông tin cần thiết, cập nhật CSDL và tổng hợp báo cáo từ dữ liệu.

Một số hệ quản trị CSDL thường gặp đó là oracle , paradox , ms access , Sybase , sql server. Và ta sẽ thực hành với hệ quản trị csdl phổ dụng đó là mysql.

Tham khảo về MySQL tại: <http://www.dbonline.jp/>

Các chức năng của hệ quản trị cơ sở dữ liệu

+ Một hệ quản trị csdl phải có khả năng giải quyết tốt những vấn đề mà cách tổ chức csdl đặt ra, hai khả năng cơ bản là:

* Quản lý dữ liệu ở mức xử lý tệp như 1 hệ điều hành
* Truy cập các khối dữ liệu lớn có hiệu quả

+ Cung cấp giao diện giữa users và csdl, giữa csdl với các hệ thống khác

+ Cung cấp 1 số ngôn ngữ bậc cao

+ Quản lý giao tác , phân quyền và an toàn dữ liệu

+ Điều khiển sự tương hợp ,tính toàn vẹn

+ Kiểm tra độ tin cậy của dữ liệu

Đa số các hệ quản trị csdl đều sử dụng 1 loại ngôn ngữ có tên là SQL để thao tác với dữ liệu. SQL là hệ thống các ký hiệu để mô tả CSDL, được hệ quản trị csdl cung cấp cho người dùng. Nó giúp ta có thể khai báo dữ liệu, khai báo các cấu truc dữ liệu thể hiện thông tin , thông báo các ràng buộc trên dữ liệu được lưu trữ trong csdl.

==================================================================

**Webserver**

**Tìm hiểu về Webservice Apache**

Tham khảo : <http://www.adminweb.jp/apache/>

Webserver được sử dụng trong Ruby đó chính là WEBRick.

WEBRick là 1 thư viện của Ruby cung cấp chức năng HTTP Webserver đơn thuần.So với apache thì nó có ít chức năng hơn nhưng từ phiển bản Ruby 1.8 trở lên, nó đã được cài đặt như 1 tiêu chuẩn trong Ruby.

Để cài đặt apache trong vagrant thì ta tham khảo tại 2 page sau:

<http://qiita.com/tamano/items/9e0c91b570bc07e3ffaf>

Đến đoạn ApacheとPassengerの設定 thì không thực hiện được nữa nên ta chuyển qua phần config file như theo link sau

<http://qiita.com/kijitoraneko/items/c5520c4e75e26176d7e1>

từ chỗ Apacheの設定, nhưng ta cần phải tạo file mới bằng vim nên sử dụng command của Gitbash thì mới có thể coppy paste được. Chú ý sử dụng 2 phím Ctr+C , Ctr+V để thiết lập chế độ coppy hoặc không.

Phần config file thì phải xem ở đây

<http://d.hatena.ne.jp/MonteCut/20111114/1321280563>

Để sửa file bằng vim ta thao tác như sau:

Ctr(phải) + V rồi ấn C là nó cho insert

Muốn thoát chế độ đó thì đặt chỏ chuột vào 1 dòng và Ctr(trái)+ C để ra khỏi chế đô insert và :wq để save + quit hoặc :q để quit

Muốn xóa file trong linux thì ta sử dụng lệnh

>rm –rf tên\_file/tên\_folder

Trong trường hợp ko được quyền xóa thì ta sử dụng quyền root

>sudo rm –rf tên\_file

Chú ý: /home và /etc là đồng cấp

>/home/vagrant/

>/etc/httpd/conf.d/ -> Tại đây thì ta sẽ khởi tạo 1 file config

|  |
| --- |
| LoadModule passenger\_module /home/vagrant/.rbenv/versions/2.2.3/lib/ruby/gems/2.2.0/gems/passenger-5.0.18/buildout/apache2/mod\_passenger.so  <IfModule mod\_passenger.c>  PassengerRoot /home/vagrant/.rbenv/versions/2.2.3/lib/ruby/gems/2.2.0/gems/passenger-5.0.18  PassengerDefaultRuby /home/vagrant/.rbenv/versions/2.2.3/bin/ruby  </IfModule> |

<http://d.hatena.ne.jp/URI/20120204/p2>

<http://qiita.com/tarosay/items/ca511ea98d4019803aee>

P/s: Vẫn chưa config được apache như theo hướng dẫn mặc dù đã config file pesenger.conf theo hướng dẫn nhưng file httpd.conf thì chưa config được vì không hiểu config ở đâu

Tìm hiểu Nginx vaf Unicorn

Nignx vaf Unicorn là gì ? Mình rất thường xuyên nghe đến 2 cái tên này, thế nhưng không hiểu chúng là gì ? Chúng làm nhiệm vụ gì ?

May mắn là mình đã tìm được 1 bài viết bằng tiếng Việt về Nginx và Unicorn. Mình sẽ nói sơ lược về 2 cái tên này tại đây.

**NGINX(エンジンエクス)**

1.Nginx là gì ?

Với câu hỏi này chắc là mình sẽ tìm được khá khá đáp án trên Google, như vây công việc của mình là đọc ,hiểu và note lại vào đây :D

**Nginx là một webserver** cực nhanh và mạnh mẽ , có tốc độ xử lý vượt trội, đặc biệt là với các file tĩnh.Nó cũng có thể được sử dụng để hoạt động như một **máy chủ “reverse proxy”** cho các giao thức HTTP , SMTP ,và **email proxy** cho POP3 và IMAP. Ngoài ra nó cũng nhằm nâng cao hiệu xuất sử lý khi sử dụng dung lượng RAM thấp.

1.1.Reverse Proxy là gì ?

======REVERSE PROXY=======

Một proxy, theo định nghĩa, là một thiết bị đứng giữa server và client, tham gia vào "cuộc trò chuyện" giữa hai bên.

Khái niệm proxy mà chúng ta thường dùng hàng ngày tốt hơn nên được gọi là một forward proxy: một thiết bị đứng giữa một client và tất cả server mà client đó muốn truy cập vào.

Reverse Proxy là gì ? Đối lập với forward proxy, đây là một loại proxy phía server. Server đóng vai trò làm reverse proxy sẽ chắn trước các request từ client đẩy đến và che dấu toàn bộ backend server đằng sau. Client sẽ không thể biết đang có một nhóm các backend server đằng sau reverse proxy phục vụ cho nó ( cái này gọi là đặc tính trong suốt của proxy. Đặc tính này giúp đơn giản hóa nhiều cho kết nối từ client. Client biết càng ít thì càng tốt )

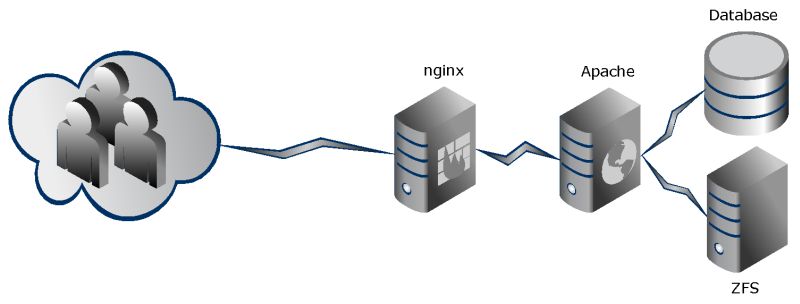
Tham khảo :

+ <http://www.stdio.vn/articles/read/175/khai-niem-ve-proxy-forward-proxy-reverse-proxy>

+ <http://www.sonand.com/2012_04_01_archive.html>

==========================

Hiện nay, ngta đang dần thay thế apache bằng nginx vì tốc độ xử lý của nó, tuy nhiên trong nhiều trường hợp thì vì một số lý do ta vẫn phải sử dụng apache. Nên ta sẽ sử dụng nginx như là 1 reverse proxy cho apache. Lúc này nginx sẽ đóng vai trò như front end xử lý các file tĩnh, còn apache sẽ làm backend xử lý dynamic content.



Tham khảo: <http://hocvps.com/cau-hinh-nginx-lam-reverse-proxy-cho-apache/>

2.Cài đặt nginx trong AWS EC2

Lệnh khởi động nginx:

>sudo /etc/init.d/nginx start

Nếu thấy FAILED thì nghĩa là cổng 80 đã được sử dụng.

Ta kiểm tra điều đó bằng cách sử dụng lệnh

>sudo yum install lsof # Cài gói lsof

>sudo lsof –i:80 # Kiểm tra cổng 80 đã có những USER nào sử dụng

Nếu có apache thì ta cần stop nó lại vì thường thì cổng 80 được mặc định sử dụng cho cả apache và nginx

>sudo apachectl stop

>sudo /etc/init.d/nginx start

Starting nginx: [OK]

Như vậy là server nginx đã được khởi động.

Giờ thì ta sẽ thử test bằng cách truy cập vào URL:

<http://public_dns:80/> hoặc <http://public_ip:80/>

VD: http://54.178.177.143:80/

Tham khảo : <http://expressmagazine.net/posts/view/3241/tin-hieu-nginx-server-can-bang-tai-unicorn-server-http-cho-ruby>

3.Các thông số của file nginx.conf

Sau khi cài đặt xong nginx thì tất cả cấu hình của nó sẽ được config trong 1 file config tại thư mục: /etc/nginx/

# /etc/nginx/nginx.conf.default

# /etc/nginx/nginx.conf

Trong file có thiết lập 1 số thông số cơ bản mà ta cần biết

Ta có thể tham khảo cấu hình config và ý nghĩa của việc cấu hình trong bài viết: <https://dongquangphuong.wordpress.com/2013/11/18/toi-uu-hoa-nginx-va-php-fpm/>

Tham khảo loạt bài viết:

+ <https://dongquangphuong.wordpress.com/tag/nginx/>

+ <http://blog.cuongnv.com/search/label/nginx> (Đây là 1 bài viết hay với ví dụ thực tế, dễ hiểu với nginx)

Vì sao nginx ko hỗ trợ .htaccess: <http://www.bloghosting.vn/tai-sao-nginx-khong-ho-tro-htaccess/>

**UNICORN**

Unicorn là server HTTP cho Ruby

Nginx gửi các request tới worker Unicorn thông qua Unix Domain Socket or TCP . Mỗi server đều có một con số worker nhất định mà trong giờ “cao điểm ” có thể đáp ứng được đồng thời nhiều yêu cầu bằng cách sắp xếp hàng đợi . Unicorn biết rõ được các worker đang xử lý tiến trình nào , hay bao bao lâu mỗi worker sử lý một yêu cầu. Thay vì xếp chồng hàng đợi chồng chất Unicorn sẽ hủy bỏ worker và ngay lập tức tạo 1 worker mới để phục vụ yêu cầu

**KVS**

**1.Memcached**

Memcached là hệ thống phân phối và lưu trữ bộ nhớ đệm, chứa dữ liệu trực tiếp trên RAM. Nó là ứng dụng mã nguồn mở, miễn phí , chủ yếu dùng để làm bộ nhớ đếm nhằm tang tốc cho ứng dụng web.

* 1. Ưu và nhược điểm của memcached

**Ưu điểm** của Memcached

+ Memcached sử dụng RAM làm nơi lưu trữ dữ liệu nên có tốc độ truy xuất rất nhanh

+ Memcached có khả năng mở rộng bằng cách liên kết nhiều server thành cụm.

+ Memcached sử dụng cấu trúc lưu trữ đơn giản key => value nên việc sử dụng và xử lý rất nhanh.

+ Memcached có thể sử dụng để lưu trữ dữ liệu SESSION cho PHP với các bước cấu hình đơn giản.

**Nhược điểm** của Memcached

+ Sử dụng RAM để lưu trữ dữ liệu nên chi phí sử dụng khá cao.

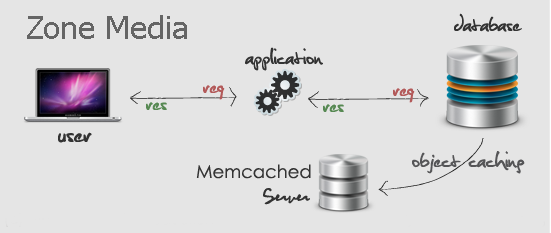
+ Nếu sử dụng Memcached trên Server/VPS khác có thể gây tác dụng ngược về hiệu năng.

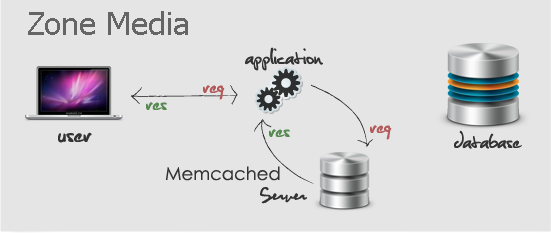
Tham khảo: <http://blog-xtraffic.pep.vn/memcached-la-gi/>

* 1. Memcached sử dụng với Nginx như thế nào ?

+ Khi một client browse đến webserver thì NGINX sẽ xử lý đầu tiên. Nếu như có cấu hình Reverse-proxy trên NGINX, thì sẽ gửi tín hiệu GET cho Memcached, hỏi thử xem trong Memcached có lưu sẵn dữ liệu không. Nếu có thì sẽ trả ngay về cho NGINX xứ lý và respond tức thời cho client.

+ Nếu như không cấu hình Reverse-proxy hoặc trong Memcached không có cache lại dữ liệu, thì lúc này NGINX sẽ gửi tín hiệu GET xuống cho các module xử lý. Một phần sẽ cache lại dữ liệu đưa vào Memcached để Memcached sử dụng cho lần sau (Nếu có cấu hình Reverse-proxy).





Mô hình làm việc được viết khá rõ trong bài viết sau:

<http://svuit.vn/web-server-database-95/memcache-va-co-che-hoat-dong-548.html>

Một bài viết quá hay về memcached: <http://codekeo.com/memcache-va-memcached/>

**2.Redis**

Một bài viết rất kĩ về Redis

<https://viblo.asia/dinhhoanglong91/posts/NznmMdXzMr69>

<https://viblo.asia/dinhhoanglong91/posts/ZK1ov1n1R5b9>

<https://viblo.asia/dinhhoanglong91/posts/jlA7GKBdMKZQ>

<https://viblo.asia/dinhhoanglong91/posts/d6BAMYDzRnjz>

2.1.Cài đăt

$ sudo yum --enablerepo=epel install redis

Tham khảo: <http://qiita.com/stoshiya/items/b8c1d2eb41770f92ffcf>

2.2.Sử dụng:

2.2.1.String

<http://nodejs.vn/topic/20/redis-string-v%C3%A0-c%C3%A1c-tr%C6%B0%E1%BB%9Dng-h%E1%BB%A3p-s%E1%BB%AD-d%E1%BB%A5ng>

2.2.2.Hashes

<http://nodejs.vn/topic/21/redis-hashes-v%C3%A0-c%C3%A1c-tr%C6%B0%E1%BB%9Dng-h%E1%BB%A3p-s%E1%BB%AD-d%E1%BB%A5ng>

2.2.3.List

<http://nodejs.vn/topic/22/redis-lists-v%C3%A0-c%C3%A1c-tr%C6%B0%E1%BB%9Dng-h%E1%BB%A3p-s%E1%BB%AD-d%E1%BB%A5ng>

2.2.4.Set

<http://nodejs.vn/topic/24/redis-sets-v%C3%A0-c%C3%A1c-tr%C6%B0%E1%BB%9Dng-h%E1%BB%A3p-s%E1%BB%AD-d%E1%BB%A5ng>

2.2.5.Sorted Sets

<http://nodejs.vn/topic/23/redis-sorted-sets-v%C3%A0-c%C3%A1c-tr%C6%B0%E1%BB%9Dng-h%E1%BB%A3p-s%E1%BB%AD-d%E1%BB%A5ng>

3.Tổng quát

Memcached thì đơn giản nhất, chỉ có dạng key-value, tất cả dữ liệu lưu trong RAM. Và lưu ý là Memcached chỉ là tầng cache, ko có tính persistent, tức là khó backup dữ liệu, và dữ liệu có thể mất (cái này là tỉ lệ eviction/reclaim xảy ra khi full bộ nhớ, memcached dùng thuật toán đẩy các key ít dùng hoặc key cũ nhất ra). Ngoài ra memcached còn có khái niệm slab, chunk, chunk grow factor, CAS v.v.

Redis thì phức tạp hơn, có nhiều kiểu dữ liệu string, hashes, sets, sorted sets, transaction, pub/sub v.v.và hình như còn cho phép scripting theo ngôn ngữ LUA (cái này mình chưa nghiên cứu). Kiểu string thì giống hết memcached, tức là key-value. Các kiểu kia có đặc trưng khác nhau. Kiểu string cho phép đặt expire time, nếu ko đặt thì key đó sẽ tồn tại mãi mãi. Redis giống memcached là memory-mapped, tức là tất cả dữ liệu đc đẩy hết vào RAM (có cách để đẩy vào swap), và cho phép theo một khoảng thời gian thì lưu một bản dump của tất cả dữ liệu vào 1 file. Redis hơn memcached ở một điểm nữa là cho phép replication. Nhưng kém memcached một điểm là hay bị dùng rất nhiều RAM. Dữ liệu ở file dump tầm khoảng 1GB, nhưng map vào RAM lên tầm 2GB

Cache là gì ?

Khi tìm kiếm trên mạng, thì bạn có thể nhận được rất nhiều câu trả lời và đa phần các tài liệu sẽ nói về bộ nhớ cache trong máy tính. Tuy nhiên hôm nay mình cần nghiên cứu về web cache nên cũng sẽ ko đi sâu tìm hiểu bộ nhớ cache đó là gì, có thể tham khảo các bài viết sau để có cái nhìn rõ hơn về cache memory

Tham khảo

+ <http://quantrimang.com/tim-hieu-cach-lam-viec-cua-bo-nho-cache-44061>

+ <http://quantrimang.com/tim-hieu-cach-lam-viec-cua-bo-nho-cache-phan-2-44116>

+ <http://quantrimang.com/tim-hieu-cach-lam-viec-cua-bo-nho-cache-phan-3-44140>

+ <http://quantrimang.com/tim-hieu-cach-lam-viec-cua-bo-nho-cache-phan-cuoi-44177>

Cache là bộ nhớ đệm dùng để lưu trữ dữ liệu tạm thời được trình duyệt web(brower) của bạn thu thập khi bạn lướt web .Các nội dung mà bộ nhớ cache lưu lại là các nội dung ít thay đổi, để trình duyệt sẽ không phải load lại các resource kích cỡ lớn(hình ảnh, file css, file js,..) sau mỗi lần truy cập của người dùng giúp cho giảm thời gian yêu cầu cũng như giảm bang thông tiêu thụ cho mỗi yêu cầu đến phía server.

Web caching là 1 kỹ thuật sử dụng cache vào hệ thống web, nó sẽ chuyển bản coppy của các tài nguyên web từ server đến gần phía client hơn.

Web Cache Server sẽ có chức năng như web server khi nhìn từ phía client và có chức năng như 1 client khi nhìn từ phía web server.Chức năng của web cache server là lưu trữ các tài liệu web(html,css,hình ảnh,..) tại vị trí gần với người sử dụng đến tránh trường hợp phải chuyển đi chuyển lại nhiều lần cùng 1 tài liệu giữa client và server, như vậy nó đã giúp giảm thời gian kết nối và giảm tải trên các server đầu ra.

Nội dung của 1 trang web thì có thể được lưu trữ ở nhiều vị trí khác nhau dọc theo các đường dẫn giữa các client và server gốc.

a.Client Cache(Brower Caches)

Đầu tiên, các trình duyệt web(brower) cũng tự xây dựng cho mình các hệ thống cache bên trong nó. Người ta gọi đó là Brower Caches.Ví dụ đơn giản đó là khi ta back thì trình duyệt đã sử dụng chính bộ nhớ cache của nó để cung cấp cho người dùng nội dung web đã được lưu trong đó, thay vì yêu cầu lại từ phía bên server gốc. Các đối tượng cache ở trong brower cache được lưu tạm thời trên đĩa hoặc trong bộ nhớ trong. Khi bạn cài đặt thì các trình duyệt web sẽ chủ động chiến lấy 10-100Mb không gian ở đĩa cứng giành cho brower caches.

Tham khảo: <http://doc.edu.vn/tai-lieu/de-tai-tim-hieu-ve-web-caching-va-phan-tich-thiet-ke-va-xay-dung-he-thong-6421/>

b.Server Cache(Proxy Caches)

**+ Proxy:** Chỉ một hệ thống Computer hoặc một Router tách biệt kết nối, giữa người gửi (Sender) và người nhận (Receiver). Nó đóng vai trò là một hệ thống chuyển tiếp (Relay) giữa 2 đối tượng: Client (muốn truy cập tài nguyên) và Server (cung cấp tài nguyên mà Client cần).

Nhờ chức năng chuyển tiếp (trung chuyển có kiểm soát) này, các hệ thống Proxy (hay Proxy servers trạm cài đặt proxy) được sử dụng để giúp ngăn chặn attacker xâm nhập vào Mạng nội bộ và các proxy cũng là một trong những công cụ được sử dụng để xây dựng Firewall trong Mạng của các tổ chức có nhu cầu truy cập Internet.

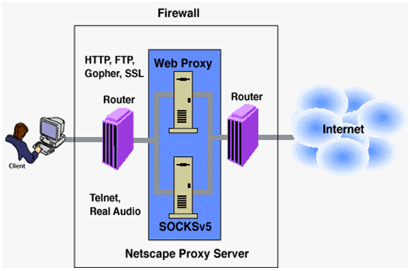
Từ proxy còn có nghĩa “hành động nhân danh một người khác” và thực sự Proxy server đã làm điều đó, nó hành động nhân danh cho Client và cả Server . Tất cả các yêu cầu từ Client ra Internet trước hết phải đến Proxy, Proxy kiểm tra xem yêu cầu nếu được cho phép, sẽ chuyển tiếp có kiểm soát yêu cầu ra Internet đến server cung cấp dịch vụ (Internet Hosts). Và cũng tương tự sẽ phản hồi (response) hoặc khởi hoạt các yêu cầu đã được kiểm tra từ Internet và chuyển yêu cầu này đến Client. Cả hai Client và Server nghĩ rằng chúng nói chuyện trực tiếp với nhau nhưng thực sự chỉ “talk” trực tiếp với Proxy.

**Tóm lại :** Proxy chỉ một hệ thống Computer hoặc một Router tách biệt kết nối, giữa người gửi (Sender) và người nhận (Receiver) proxy có địa chỉ IP và một cổng truy cập cố định.(tất nhiên là phải khác nhau theo từng địa phương và từng nước)

**+ Proxy Server:** Proxy Server là một server đóng vai trò cài đặt proxy làm trung gian giữa người dùng trạm( workstation user) và Internet. Với Proxy Server, các máy khách( clients) tạo ra các kết nối đến các địa chỉ mạng một cách gián tiếp. Những chương trình client của người sử dụng sẽ qua trung gian proxy server thay thế cho server thật sự mà người sử dụng cần giao tiếp.

Proxy server xác định những yêu cầu từ client và quyết định đáp ứng hay không đáp ứng, nếu yêu cầu được đáp ứng, proxy server sẽ kết nối với server thật thay cho client và tiếp tục chuyển tiếp đến những yêu cầu từ client đến server, cũng như đáp ứng những yêu cầu của server đến client. Vì vậy proxy server giống cầu nối trung gian giữa server và client.

Hiểu một cách đơn giản là : Proxy server là một trung tâm cài đặt các proxy .Mà các proxy này nằm giữa máy tính của bạn và tài nguyên internet (bộ đệm) mà bạn đang truy nhập . Dữ liệu mà bạn yêu cầu đến proxy trước , rồi sau đó nó mới truyền dữ liệu cho bạn và ngược lại.



Tại sao lại dùng Proxy ?

* Do mọi thông tin truy xuất phải thông qua Proxy nên chúng ta có thể quản lý được mọi thông tin ra và vào ví dụ: Mọi yêu cầu của máy khách phải qua Proxy server, nếu địa chỉ IP có trên proxy, nghĩa là website này được lưu trữ cục bộ, trang này sẽ được truy cập mà không cần phải kết nối Internet, nếu không có trên Proxy server và trang này không bị cấm, yêu cầu sẽ được chuyển đến server thật, DNS server… và ra Internet.
* Các dịch vụ proxy đều có lợi trong việc logging :Vì các proxy server hiểu các giao thức cơ bản, chúng cho phép logging đạt hiệu quả. Ví dụ, thay vì logging tất cả những dữ liệu đã truyền, một FTP (File Transfer Protocol) proxy server chỉ ghi lại những lệnh đã tạo và những đáp ứng của remote server, điều này giúp việc logging ít và hữu dụng hơn.
* Đáp ứng được nhu cầu truy xuất của cá nhân và vừa đảm bảo an toàn cho hệ thống cục bộ do chúng ta sử dụng địa chỉ ẩn danh ,và mọi truy xuất đều thông qua proxy nên thông tin cục bộ không trực tiếp tương tác với bên ngoài.
* Các dịch vụ proxy cho phép người dùng truy cập các dịch vụ Internet “trực tiếp”. Với các dịch vụ Proxy, các người dùng luôn nghĩ rằng họ đang tương tác trực tiếp với các dịch vụ Internet. Ví dụ các người dùng chỉ cần gõ vào địa chỉ của một trang web nào đó thì trang web được trình duyệt hiển thị lên cho người dùng. Dĩ nhiên là có nhiều công việc phải làm ở bên trong nhưng nó là trong suốt đối với người dùng. Người dùng truy cập các dịch vụ Internet từ chính những hệ thống riêng của họ, mà không cần cho phép các gói tin truyền trực tiếp giữa hệ thống của người dùng và Internet đảm bảo an toàn cho hệ thống.
* Proxy server tích lũy và cứu file , những file mà thường đựơc yêu cầu bởi ngàn người dùng trên internet trong dữ liệu đặc biệt , gọi là cache . Do đó , proxy server chúng có thể tăng tốc độ truy nhập internet. Cache của proxy server có thể đã sẵn chứa thông tin bạn cần trong thời gian bạn yêu cầu , làm cho proxy server có thể phân phối thông tin ngay lập tức mà không cần phải truy tìm thông tin ngoài internet.
* Một Proxy Server thường nằm bên trong tường lửa , giữa trình duyệt web và server thật , làm chức năng tạm giữ những yêu cầu Internet của các máy khách để chúng không giao tiếp trực tiếp Internet .Người dùng sẽ không truy cập được những trang web không cho phép ( bị công ty cấm ). Vd :Admin không muốn nhân viên của mình đọc báo hay chơi game online trong giờ làm việc , bằng cách dùng proxy server admin có thể khóa một số site được chỉ định.
* Proxy server làm cho việc sử dụng băng thông có hiệu quả do chúng ta quản lý được các hoạt động của người dùng.Nên có thể giới hạn thông tin nào được dùng và không dùng tránh được việc nghẽn băng thông.

Bây giờ chúng ta sẽ đi sâu hơn về các tính năng chính của proxy Server để hiểu vì sao lại dùng Proxy Server.

Các tính năng chính của Proxy server gồm 3 chức năng chính

Tường lửa và filtering.

Chia sẻ kết nối.

Caching.

Không giống như brower caches chỉ phục cho 1 đối tượng người dùng thì proxy caches có thể phục vụ nhiều users 1 lúc. Khi có nhiều người cùng truy cập vào 1 website thì cache proxies thường được sử dụng hơn là brower caches

Tham khảo: <http://vdo.vn/may-chu-proxy-proxy-server>

Tham khảo:

+ <http://thietkeweb24.info/blog/kien-thuc-co-ban/cache-cookie-session-view-state-la-gi_16.html>

+ <http://tek.eten.vn/toi-uu-hieu-nang-website-cache>

+ <http://www.nguyentrongtho.com/2013/10/Toi-uu-toc-do-tai-trang-web-http-cache.html>

+ <https://bienuit.wordpress.com/tag/cache-la-gi/>

+ <http://doc.edu.vn/tai-lieu/de-tai-tim-hieu-ve-web-caching-va-phan-tich-thiet-ke-va-xay-dung-he-thong-6421/>