Big News

骆神 wrote in the News Section:

喜迎校庆 SJTUG Mirrors加入Arch Linux镜像源

在上海交通大学120周年校庆前夕，经过SJTUG所有人的共同努力，我们上线了Arch Linux，PuTTY和Cygwin的镜像，向百廿交大的生日献上我们的祝福。

Arch Linux源使用方法:

编辑/etc/pacman.d/mirrorlist，先注释掉里面的所有行，然后在文件的最顶端添加

```
Server = http://mirrors.sjtug.org/archlinux/$repo/os/$arch
```

~~How to make a big news~~ SJTUG Mirrors

Lv Zheng

SJTUG

lv.zheng.2015@gmail.com

April 23, 2016

Questions to Ask

Q0

How great is the server's hardware?

Q1

How does a mirror work?

Q2

How do we deploy the services?

How great is the server's hardware?

Let's have a look!

How does a mirror work?

tunasync

Different websites/repos require different tools to synchronize, and the most used tool for synchronization is `rsync`. However, some sites such as *Debian archive* requires 2-stage sync, while some others don't.

`tunasync` is a great tool for such operations. It supports different sync protocols. Sites to sync are configured in a config file. For example, to sync PuTTY (which is a very small site, less than 100 MB in size), add these lines to the config file:

tunasync.conf

```
[[mirrors]]
name = "PuTTY"
provider = "rsync"
upstream = "rsync://rsync.chiark.greenend.org.uk/\
ftp/users/sgtatham/putty-website-mirror/"
```

~~Too young~~ too simple, isn't it?

nginx

`nginx` is a even faster HTTP server program than ~~western~~ Apache HTTPD, especially for serving static contents, thus very suitable for our mirrors. However, dynamic contents such as PHP generated pages can be served with FastCGI.

How do we deploy the services?

Traditional way (1)

Install all the required software packages such as Apache, MariaDB, and PHP on the host and run them on the host space.

Pros:

- .....

Cons:

- One security flaw of a software may break the entire system. (may be avoided by using a non-privileged user for each service)
- Hard to deploy, especially hard for cluster deployment.
- Hard to deploy multiple services that use the same software.
- Hard to make backups.

Traditional way (2)

Virtually same as the previous way. Just do everything in a virtual machine instead.

Pros:

- Easy for cluster deployment and backups.
- May deploy multiple services on a single computer.
- Strong isolation.

Cons:

- Performance downgrade. Low memory-efficiency. Cache unfriendly.
- It's excited~~ed~~ing to do so on a cloud server :)

The Docker way

What is Docker

- Docker allows you to package an application with all of its dependencies into a standardized unit for software development.
- Docker containers wrap up a piece of software in a complete filesystem that contains everything it needs to run: code, runtime, system tools, system libraries - anything you can install on a server. This guarantees that it will always run the same, regardless of the environment it is running in.

(凑表脸地偷懒了)

Let's learn Docker from an example

批判一番

Rewrite the whole project thoroughly

Having learned about the `mirror-docker` project, I decided to rewrite the `Dockerfiles` and scripts.

With our joint efforts, the project was finally rewritten and put into application.

We added `archlinux`, `PuTTY`, and `cygwin` repos right before SJTU 120 Anniversary.

Structure of the new settings

Currently there are two Docker containers running on the server, and another container running on demand.

1. `tunasync` container syncs the repos and generate `tunasync.json` regularly. Files are stored in a host directory mounted to the container.
2. `nginx` container serves the HTTP content. On each request, it first looks up the `site` directory for web content. If not found, serve the `tunasynced` files.
3. `jekyll` container builds the static web pages from source. It is manually run on demand, for example, when news are updated.

識得唔識得啊

Sik Dak Ng Sik Dak Aa

Who is to blame

- 骆神 ~~gives us a negative example of `Dockerfile`,~~ 钦定 the entire project, and 骗来服务器。
- 李臻博 ~~violenty~~ upgrades docker on the server and contributes a lot of precious ideas and script code to the rewrite.
- 吕正 does ~~destructive~~ experiments and deploys the services on the server.
- And thank you all for testing the mirrors.

What Next?

0.1 Milestone:

1. System monitor
2. HTTPS support

1.0 Milestone:

1. Rewrite the Web frontend

Also,

1. Establish a working group for mirrors.
2. Find a better way to manage privileges on the server
3. Obtain more computing resources, especially disk space and bandwidth

The End

Terribly sorry for broken English