

# eAPT: enhancing APT with a mirror site resolver

Gilhee Lee  
Sungkyunkwan University  
Suwon, South Korea  
2gilhee@skku.edu

Taegeun Moon  
Sungkyunkwan University  
Suwon, South Korea  
taegeun@skku.edu

Min Jang  
Samsung Electronics  
Suwon, South Korea  
min51th.jang@samsung.com

Hyounghick Kim  
Data61, CSIRO  
Sydney, Australia  
Hyounghick.Kim@data61.csiro.au

**Abstract**—Advanced Packaging Tool (APT) is a package manager used in Debian Linux distributions. By default, APT is configured to use the official central repository, but it also allows users to modify and add alternative repositories easily. Since using geographically closer servers can boost the download speed, many users adopt mirror sites in their country instead of the default repository. However, it is often challenging for users to select an appropriate mirror site. First, users have to change the repository URL manually with insufficient information about mirror sites. Second, if the mirror site in use is not working, APT cannot find an alternative mirror site in an automated manner. To address these problems and improve the user experience of APT, we propose the enhanced Advanced Packaging Tool (eAPT) which is built on the top of APT with a mirror site resolver to find the optimal mirror site based on the user’s geographical location in terms of the package installation time and stability even when some mirror sites are not responsible. Our experimental results demonstrate that eAPT is about between 8 and 10 times faster than APT for installing large sized packages (e.g., `openjdk-11-jdk` or `android-sdk`) in India and Australia.

**Index Terms**—APT, package manager, server selection

## I. INTRODUCTION

A package manager is a collection of software tools that automate the process of installing, upgrading, managing and removing programs for operating systems (OSes). Previously, it is a quite time-consuming and cumbersome task to install and update programs without a package manager because users had to download packages manually from the internet, which occasionally require manual configurations before installation. However, with the package manager’s use, such annoying tasks were automated, thus alleviating the user’s burden. In addition, the package manager increased the convenience of installation and removal of programs, version update, and installing dependencies. Therefore, a package manager has been recognized as an essential component of the operating system. The state-of-the-art operating-system provides its package manager. Table I shows various package managers for each operating system most popularly used.

Operating system	Debian	Redhat	Fedora	OSX
Package manager	<i>apt</i>	<i>rpm &amp; yum</i>	<i>yum &amp; dnf</i>	<i>brew*</i>

\* : unofficial, but widely used

TABLE I: Package managers for Unix family of operating systems.

Most of the package managers, including APT, has an official central package repository that serves package files.

For example, Ubuntu maintains a central repository web site (<http://archive.ubuntu.com/ubuntu>). Unsurprisingly, the package download speed depends on the geographical distance between the server and the client. We empirically found that the worst-case client’s download speed can become slower about four times than the best-case client’s download speed. Several optimization techniques, such as ISP caching, Content Delivery Network [1] and GeoDNS [2], have been introduced to address this issue. However, the use of mirror sites [3] has become most popular in practice because it does not require any modification of the existing internet infrastructure.

APT allows users to modify and add other repositories, which are distributed all over the world. These alternative repositories are generally referred to as *mirror sites* since they maintain a copy of the official repository. Many voluntary organizations and individuals typically maintain mirror sites. With the mirror sites, users can use APT conveniently, even if they are geographically far away from the official repository.

However, even though mirror sites solved the latency problem due to the geographical distance, there remains a problem of selecting the optimal mirror sites. A typical selection method is to choose a mirror site among the mirror sites in a list such as “<http://mirrors.ubuntu.com>” [4] and “<https://launchpad.net/ubuntu/+archivemirrors>” [5]. In general, it is quite difficult for casual users to manually select an optimal mirror site to reduce the download speed because such a list only provides the country information of each mirror site. Moreover, if the selected mirror site is not working, it also requires changing it to another manually. Finally, there is no load balancing mechanism among mirror sites – many users would be concentrated on some mirror sites.

To solve those problems, we propose an APT management scheme called enhanced APT (eAPT) built on the top of APT with a mirror site resolver to find the optimal mirror sites for Ubuntu users. eAPT leverages a list of mirror sites at the country-level and selects the top  $k$  geographically closest mirror sites as *candidate mirror sites*. Whenever a user tries to install packages, eAPT measures the latency of all candidate mirror sites and selects the one with the lowest latency. Our experimental results demonstrate that eAPT allows users to download packages in an efficient manner.

Our contributions are summarized as follows:

- We develop a practical package management framework called eAPT that finds an optimal mirror site for Ubuntu users to always maximize the download speed and reli-

ability. Our code can be found at a GitHub repository (<https://github.com/eAPT2020/eAPT>).

- We perform experiments in six regions (South Korea, USA, UK, Germany, India, and Australia) to show the superiority of eAPT over APT. Our experimental results demonstrate that eAPT is about between 8 and 10 times faster than APT for installing large sized packages (e.g., `openjdk-11-jdk` or `android-sdk`) in India and Australia.

## II. BACKGROUND

### A. Advanced Packaging Tool

Advanced Packaging Tool (APT) is a collection of package manager tools used on Debian family Linux distros, including Ubuntu. It is widely used to install and upgrade `.deb` packages and their dependencies.

In the Ubuntu image, the URL of APT is set to `http://archive.ubuntu.com/ubuntu/` or `http://<country-code>.archive.ubuntu.com/ubuntu/` by default. The repository configuration file of APT is located in `/etc/apt/sources.list`. Users can update the list of available packages from the repositories specified in `/etc/apt/sources.list` using the `apt update` command. After the update, APT will cache the list of available packages in `/var/lib/apt/lists` (see Fig. 1). Users can selectively download and install tools from the site using the `apt install` command.

```
apt.kubernetes.io_dists_kubernetes-xenial_InRelease
apt.kubernetes.io_dists_kubernetes-xenial_main_binary-amd64_Packages
dl.bintray.com_sbt_debian_Packages
dl.bintray.com_sbt_debian_Release
dl.bintray.com_sbt_debian_Release.gpg
dl.yarnpkg.com_debian_dists_stable_InRelease
dl.yarnpkg.com_debian_dists_stable_main_binary-all_Packages
dl.yarnpkg.com_debian_dists_stable_main_binary-amd64_Packages
dl.yarnpkg.com_debian_dists_stable_main_binary-i386_Packages
download.sublimetext.com_apt_stable_InRelease
download.sublimetext.com_apt_stable_Packages
ftp.daumkakao.com_ubuntu_dists_xenial-backports_InRelease
ftp.daumkakao.com_ubuntu_dists_xenial-backports_main_binary-amd64_Packages
```

Fig. 1: List of available packages in `/var/lib/apt/lists`.

### B. Mirror Site

A mirror site is a replica of a network site located in a different geographical place. The purpose of a mirror site is to provide the same data and services as the original site faster and more efficiently for geographically distant users. Some mirror sites are officially provided, and some are operated by individuals, companies, etc. Most mirror sites exist in each country, and users can manually find and update their mirror sites.

To keep the consistency with the original site, mirror sites periodically synchronize with the original site. For example, the Kakao server (<http://mirror.kakao.com>), which is the most popular mirror site in South Korea, synchronizes the package lists with the original site every six hours.

### C. Problems with Mirror Sites

The existing APT scheme with mirror sites raises the following two problems.

1) *Manual configuration efforts*: By default, the official package repository (<http://archive.ubuntu.com/ubuntu/>) is set as the server for updating and installing the packages for Ubuntu. Therefore, users who are geographically distant from the site might have the disadvantage of installing and updating packages. As mentioned in Section II-B, the use of mirror sites can be useful for such users. However, users should manually find a mirror site and configure it. Perhaps, it would be a not easy task for casual users. Furthermore, since the list of mirror sites only shows which country each site belongs to, users in a vast country like the USA still have to select their mirror site carefully.

Domain	A records
kr.archive.ubuntu.com	91.189.91.39
	91.189.91.38
	91.189.88.142
	91.189.88.152
uk.archive.ubuntu.com	91.189.88.142
	91.189.88.152
	91.189.91.39
	91.189.91.38
archive.ubuntu.com	91.189.88.142
	91.189.88.152

TABLE II: DNS records for the official Ubuntu repositories.

Moreover, although there exist the official mirrors sites for some countries such as USA (`us.archive.ubuntu.com`) and UK (`uk.archive.ubuntu.com`), we found that they are sometimes not the optimal mirror site in terms of network latency. The DNS lookup results for the official Ubuntu repositories are shown in Table II. Interestingly, the DNS records for `kr.archive.ubuntu.com`, which has a country prefix of South Korea, have the same records with `uk.archive.ubuntu.com`, which has a country prefix of UK, indicating that it is challenging to select a geographically closer mirror site with domain names only.

2) *No fault tolerance scheme*: Mirror sites can sometimes change their URL, or the service itself can be not responsible. For example, a mirror site's URL (<http://ftp.daum.net>) was changed twice such as <http://ftp.daumkakao.com> and <http://mirror.kakao.com> sequentially. Another mirror site (<http://ftp.kaist.ac.kr>) stopped for several months due to a server error. When such unexpected events occur, users who were using such URLs can no longer use APT. Moreover, users should manually select an alternative mirror site and configure it.

## III. ENHANCED APT (eAPT)

We implemented eAPT as a wrapper of APT in Python3. The process of finding the optimal mirror site consists of two phases: (1) *candidates selection* and (2) *optimal mirror selection*. In the *candidates selection* phase, eAPT selects the  $k$  geographically closest mirror sites because those mirror sites are likely to offer low-latency access to files. In the *optimal mirror selection* phase, eAPT selects mirror sites with the lowest latency among those  $k$  mirror sites selected from the *candidates selection* phase. Fig. 2 shows the overall process of eAPT.

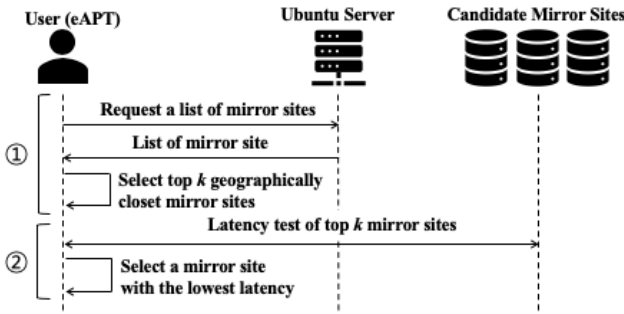


Fig. 2: Overall process of eAPT. ① Phase 1: Candidate Selection, ② Phase 2: Optimal Mirror Selection

### A. Phase 1: Candidate Selection

The official list of Ubuntu mirror sites [5] provides to which country each mirror site belongs. However, the country-level location information may not be sufficient to decide which mirror site is suitable for users. For example, there exist over 80 mirror sites in USA only. In such scenarios, a possible option is to measure all mirror sites’ latency in a country and provide the measured latency information as a useful indicator for choosing the optimal mirror site. However, this measuring practice with all mirror sites would incur a significant overhead. To minimize the overhead for measuring each site’s latency, eAPT first identifies the top  $k$  geographically closest mirror sites within the country. Fig. 2-① illustrates the process of *candidates selection*.

eAPT obtains the list of mirror sites by the user’s location from the official Ubuntu server. To find mirror sites in the same country with users, we leveraged the file (<http://mirrors.ubuntu.com/mirrors.txt>) [6] hosted by Ubuntu. Whenever a user accesses this file, it automatically redirects the file containing the mirror site list within the country where the user is accessing. Fig. 3 shows a list of the mirror sites in South Korea.

```

http://ftp.lanet.kr/ubuntu/
http://mirror.kakao.com/ubuntu/
https://mirror.misakamikoto.network/ubuntu-ports/
http://ftp.daum.net/ubuntu/
http://ftp.harukasan.org/ubuntu/
https://mirror.misakamikoto.network/ubuntu/
http://mirror.yongbok.net/ubuntu/
http://archive.ubuntu.com/ubuntu/
  
```

Fig. 3: List of mirror sites in South Korea.

Next, we used the IP Geolocation API [7] to translate the IP addresses of mirror sites into their geolocations. We calculate the Haversine distance between each mirror site and the user with their geolocations (i.e., longitude and latitude). Finally, we select the top  $k$  geographically closest mirror sites from the user. We believe that a geographically closer server from the user can provide low latency file access to the server. Mirror sites selected in this phase are called *candidate mirror sites*

(or simply “candidates”). Next, eAPT downloads the list of available packages for all candidate mirror sites and internally maintains them for the optimal mirror site selection.

### B. Phase 2: Optimal Mirror Selection

After Phase 1, eAPT provides the same interfaces with APT. Since eAPT is a wrapper of APT, it redirects all commands to APT. However, before executing the commands, eAPT measures the latency of the *candidate mirror sites* and sets the mirror site’s URL to a mirror site showing the lowest latency (see Fig. 2-②). The official documentation of Debian is recommending to use download programs such as `wget` [8] or `rsync` [9] for measurement [10]. Similarly, eAPT measures the latency of each mirror site by downloading their main web page. Specifically, eAPT uses an asynchronous programming API in Python3 to concurrently measure the download time, and uses `PyCurl`, a Python binding of `libcurl`, to measure the total download time. Finally, eAPT selects the mirror site with the lowest latency as the optimal mirror site. With the optimal mirror site information, eAPT updates the mirror site configuration in `/etc/apt/sources.list`. Furthermore, eAPT exposes only the optimal mirror site’s package list to APT.

## IV. EXPERIMENTS

To show the efficiency of eAPT for downloading packages, we evaluate the performance of eAPT compared with the conventional APT scheme.

### A. Experimental Environment

To generalize our performance measurement, we used Amazon EC2 instances in six regions: Seoul (South Korea), California (USA), London (UK), Frankfurt (Germany), Mumbai (India), and Sydney (Australia). We used `t2.micro` instances equipped with Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz, 1GB memory, and 8GB storage on Ubuntu 18.04 LTS.

To evaluate the performance overhead of finding the fastest mirror and improvement of download speed by using eAPT, we measured the execution times of APT and eAPT, including the download time and the installation time for packages.

For APT, we configured the mirror site to the official repository (<http://archive.ubuntu.com>). To reduce the bias in reported estimates, we repeated the measurements ten times for each package and took the average time. We chose 6 popularly used packages with various sizes because the package download speed can be changed with package size. The packages are categorized into three groups: small size (`tree`, `curl`), medium size (`apache2`, `mysql-server`), and large size (`openjdk-11-jdk`, `android-sdk`). The package size was calculated by summing up packages with important dependencies by using `apt-cache show` from `apt` commands. Table III shows the size of each package.

### B. Validity of Distance-based Candidate Mirror Sites Selection

To confirm that the idea of using geolocations of mirror sites is useful, we first measure network latency of all 81 mirror

package	size (MB)	package	size (MB)
tree	0.03	mysql-server	3.04
curl	0.15	openjdk-11-jdk	74.52
apache2	1.34	android-sdk	74.58

TABLE III: Packages used in the experiments.

sites in the USA and their distances from our Amazon EC2 VM instance in California and analyze the correlation between them. Fig. 4 shows that the Spearman correlation coefficient is 0.63, indicating a moderate positive relationship between them.

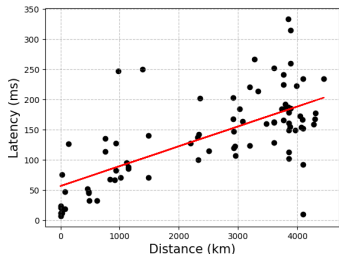


Fig. 4: Correlation between distance and latency. The Spearman correlation coefficient is 0.63.

### C. Finding the Optimal $k$

As mentioned in Section III-A, eAPT selects the top  $k$  geographically closest mirror sites for the latency test when it is first launched. Therefore, selecting the optimal  $k$  would be the most crucial design issue in deploying eAPT so that it can ensure the inclusion of the best mirror site while minimizing the latency test overhead. We performed the experiments to find the optimal  $k$ .

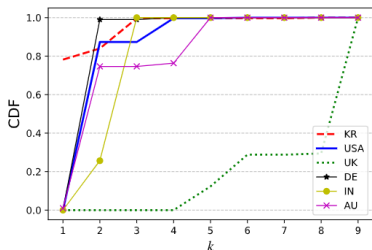


Fig. 5: Ratio of the inclusion of the best mirror site within the  $k$  closest mirror sites.

We ran eAPT with  $k=9$  every minute for six hours. The numbers of collected samples are 312, 300, 316, 300, 300, and 300, respectively, for South Korea, USA, UK, Germany, India, and Australia.

To find a necessary  $k$  covering the best mirror site, we calculate the ratio of the inclusion of the best mirror site within the  $k$  closest mirror sites from the collected samples. Fig. 5 shows the cumulative relative frequency (CDF) of that ratio, indicating that  $k$  should be large as at least 2 for Germany, 3

for South Korea and India, 4 for USA and India, and 5 for Australia. However, for UK,  $k$  should be large as at least 9 to cover the best mirror site. We surmise that this is because Ubuntu’s official server (<http://archive.ubuntu.com/ubuntu/>) is located in UK. Therefore, in the case of UK, the geolocation of the mirror site may not be an essential factor in determining the best latency mirror site.

Also, we need to select  $k$  as small as possible to reduce the latency test overhead for finding the best mirror site. Therefore, we measured how much overhead the optimal mirror selection process introduces with varying  $k$  from 3 to 9. We denote the region R’s result with  $k$  as R- $k$ . As shown in Fig. 6, for all six regions, the measurement time increases with  $k$ . In South Korea, KR-3 and KR-4 show similar measurement times, while the measurement time linearly increases when  $k > 4$  (see Fig. 6 (a)). In USA, the measurement time remains similar  $k \leq 4$ , while the measurement time linearly increases when  $k > 5$  (see Fig. 6 (b)). In UK, the measurement times are quite stable regardless of  $k$  (see Fig. 6 (c)). In Germany and India, they increase rapidly when  $k = 4$  (see Fig. 6 (d) and (e)). We note that there are only 7 mirror sites in India. In Australia, AU-3, AU-4 and AU-5 show similar measurement times, while the measurement time linearly increases when  $k > 5$  (see Fig. 6 (f)).

Therefore, to choose the optimal mirror site with the least overhead, we recommend that  $k$  should be 3. However, in the following experiments, we commonly set  $k = 5$  for all regions to ensure the inclusion of the best mirror site in a more conservative manner.

### D. Experimental Results

Fig. 7 shows the execution time results of APT and eAPT in South Korea, USA, UK, Germany, India, and Australia, respectively, on Amazon EC2. eAPT outperformed APT except UK and Germany when the size of packages is larger than about 1.3MB (i.e., `apache2`, `mysql-server`, `openjdk-11-jdk` and `android-sdk`). Notably, eAPT is more highly efficient than APT for large sized packages over 70MB (i.e., `openjdk-11-jdk` and `android-sdk`). In India and Australia, the performance improvement of eAPT is more clearly shown compared with other regions – eAPT is about between 8 and 10 times faster than APT, as shown in Fig. 7 (e) and (f).

In contrast, APT overall shows better performance than eAPT for small-sized packages such as `tree` and `curl` except in India. We found that eAPT took more than about four seconds, even with small-sized packages. We surmise this is due to the overhead of the optimal mirror selection process. If the benefit of reducing the package download time cannot compensate for the overhead caused by the optimal mirror selection process (i.e., when the size of a package is small), eAPT would not be a recommendable option.

Unlike our expectation, APT is always faster than the proposed eAPT in UK and Germany. Again, this is because Ubuntu’s official server (<http://archive.ubuntu.com/ubuntu/>) is

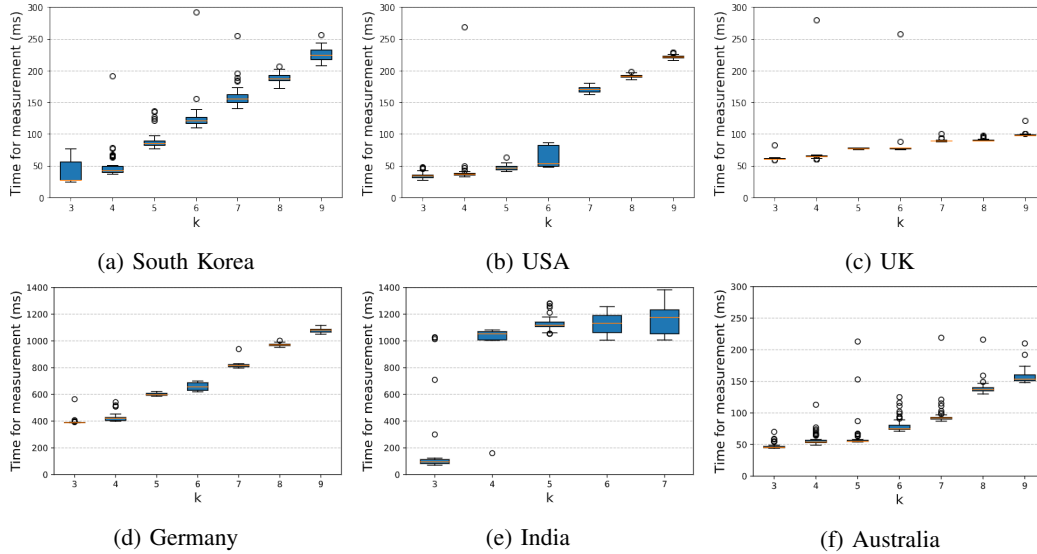


Fig. 6: Time for latency measurement on varying  $k$ .

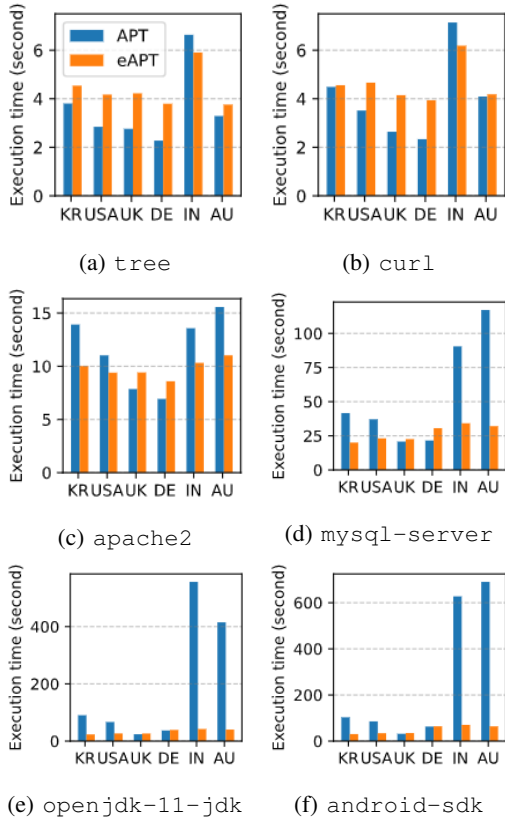


Fig. 7: Execution times for installing six popular packages.

located in UK. Therefore, it is not easy to have the benefit of using a mirror site instead of the official server.

## V. DISCUSSION

This section discusses some possible optimization techniques to improve the performance of eAPT and the limitations

of the current eAPT implementation.

### A. Optimization Strategies

In our experiments, we configured eAPT to find the optimal mirror site whenever it is executed based on the assumption that the performance of mirror sites changes dynamically. Therefore, the latency measurement incurring a significant time overhead is always performed. However, in a real-world environment, several heuristics can additionally be applied to avoid this overhead. A possible approach is to test the currently defined mirror site's latency first and then perform the optimal mirror site selection process only when the current mirror site's latency is too high, or it is unavailable. Another approach is to perform the optimal mirror selection process periodically at a specific time interval (e.g., a day) instead of performing it every time. Furthermore, the optimal mirror site selection process can be executed as a background process after the APT command execution finishes. These techniques would help reduce the overhead of the mirror selection process.

### B. Limitations

The current implementation of eAPT is not a complete solution. We still need to consider some practical issues to commercialize eAPT as a complete software bundle. For example, eAPT used the official Ubuntu site to obtain the list of mirror sites (i.e., <http://mirrors.ubuntu.com/mirrors.txt>) by country. Therefore, if the official Ubuntu site is not working or the information about this list is not correct, eAPT would fail to find a suitable mirror site. To address this problem, we can deploy backup servers providing the list of available mirror sites to make the eAPT service more reliable.

We have a few limitations in our experiments. The six regions tested may not be sufficient to generalize our experiments' observations. For example, in a small country, a server in a neighboring country might be a better option

than a server in the same country. Moreover, our experiments were performed in a cloud environment with high network bandwidth. Therefore, our experiment results may not reflect client situations under poor network connectivity.

## VI. RELATED WORK

In this section, we briefly reviewed previous work related to eAPT.

Kwon et al. [11] proposed a mechanism to find an optimal cloud server by measuring the network latency from each cloud server. They defined the latency as the time it takes to receive an acknowledgment message for the last packet after sending the first packet. Unlike their approach, we measure the network latency from mirror sites using the elapsed time to download a web page to reflect the actual throughput of each mirror site more accurately. Wendell et al. [12] presented a system called DONAR, which is an Internet-scale system to recommend the optimal replica for clients using the clients' and servers' geolocations and servers' workloads via intermediate brokers called mapping nodes. Hajjat et al. [13] proposed a system called Dealer to minimize user response times. Dealer monitors the performance of individual replicas and communication latency between replica pairs to select the best combination of cloud replicas (potentially located across different data-centers) for serving user requests. While DONAR and Dealer use a complicated architecture for balancing servers' workloads, eAPT is designed more straightforwardly to find the fastest mirror site for the package management service.

Even though package management is a practically important service, to the best of our knowledge, just one prior research attempted to develop a practical tool called fastmirror (<https://wiki.centos.org/PackageManagement/Yum/FastestMirror>) to find the fastest mirror site. Unlike eAPT built on the top of APT, fastmirror is used for yum [14], which is the package manager for the Fedora Linux distribution. fastmirror was developed as part of Duke University's project. fastmirror measures the time that takes to look up the DNS name and open a TCP connection to the mirror while eAPT measures the time to download the main pages of mirror sites to reflect the actual download time for installing packages. Moreover, unlike eAPT, which automatically selects candidates based on all the mirror sites provided by Ubuntu, fastmirror selects the best mirror site among the only mirror sites manually configured by users.

## VII. CONCLUSIONS

Ubuntu users mainly use APT to install packages selectively. For reliability and efficiency, it is common to use mirror sites that are geographically closer to users instead of the official site. However, it would be cumbersome for users to manually determine which mirror site is the fastest and most reliable one. Therefore, we propose eAPT to recommend the optimal mirror site for users in an automated manner.

We assume that servers with close geographical distances have the lowest latency. Under this assumption, eAPT first cal-

culates (approximated) geographical distances between users and mirror sites. When running eAPT first time, among the list of mirror sites located in the user's country, five nearest mirror sites are selected as candidates. For the package installation, the fastest mirror site is determined among the candidate mirror sites in an automated manner.

Our experimental results showed that eAPT outperformed the conventional APT scheme in terms of the execution time to install packages. When the size of the installed packages is larger than 70MB, there is a noticeable improvement in the execution time. For example, in India and Australia, the execution times of eAPT for `android-sdk` and `openjdk-11-jdk` are about between 8 and 10 times faster than APT.

Although our current implementation is focused on APT, we believe that the design and implementation techniques of eAPT can also be applied to other package management tools, such as PyPI (Python Package Index). Future work might try to design a generic data discovery model or algorithm for the package management problem with mirror sites.

## VIII. ACKNOWLEDGEMENTS

This work was supported by the MSIT, Korea (IITP-2020-2015-0-00742 and No. 2019-0-01343) supervised by the IITP. Note that Hyounghick Kim is the corresponding author.

## REFERENCES

- [1] "What does CDN stand for? CDN Definition," (Accessed on June. 5, 2020). [Online]. Available: <https://www.akamai.com/us/en/cdn/what-is-a-cdn.jsp>
- [2] "Use DNS Policy for Geo-Location Based Traffic Management with Primary Servers," (Accessed on June. 5, 2020). [Online]. Available: <https://docs.microsoft.com/en-us/windows-server/networking/dns/deploy/primary-geo-locations>
- [3] J. W. Byers, M. Luby, and M. Mitzenmacher, "Accessing multiple mirror sites in parallel: using Tornado codes to speed up downloads," in *Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 1999)*, 1999.
- [4] "Official list of mirror sites by country code," (Accessed on June. 5, 2020). [Online]. Available: <http://mirrors.ubuntu.com>
- [5] "Official Archive Mirrors for Ubuntu," (Accessed on June. 5, 2020). [Online]. Available: <https://launchpad.net/ubuntu/+archivemirrors>
- [6] "User location based Ubuntu mirror list," (Accessed on June. 5, 2020). [Online]. Available: <http://mirrors.ubuntu.com/mirrors.txt>
- [7] "ipstack - Free IP Geolocation API," (Accessed on June. 6, 2020). [Online]. Available: <https://ipstack.com>
- [8] "GNU Wget," (Accessed on July. 17, 2020). [Online]. Available: <http://www.gnu.org/software/wget/>
- [9] "rsync," (Accessed on July. 17, 2020). [Online]. Available: <https://rsync.samba.org>
- [10] "Debian worldwide mirror sites," (Accessed on July. 17, 2020). [Online]. Available: <https://www.debian.org/mirror/list.en.html>
- [11] M. Kwon, Z. Dou, W. Heintzelman, T. Soyata, H. Ba, and J. Shi, "Use of Network Latency Profiling and Redundancy for Cloud Server Selection," in *Proceedings of 7th IEEE International Conference on Cloud Computing*, 2014.
- [12] P. Wendell, J. W. Jiang, M. J. Freedman, and J. Rexford, "DONAR: Decentralized Server Selection for Cloud Services," in *Proceedings of the ACM SIGCOMM Conference*, 2010.
- [13] M. Hajjat, S. P. N. D. Maltz, S. Rao, and K. Sripanidkulchai, "Dealer: Application-Aware Request Splitting for Interactive Cloud Applications," in *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*, 2012.
- [14] "Yum Package Manager," (Accessed on June. 5, 2020). [Online]. Available: <http://yum.baseurl.org>