Reproduction Report of Watchman

1. Visiting records of Watchman

Recently, we promoted `Watchman` in the open source community by communicating with developers in the comments section of issue reports submitted by us. We also invited several experienced developers from two Chinese software companies to apply `Watchman` to their company projects. For research purposes, we recorded the how visitors interacted with `Watchman` homepage from 9 December, 2019 to 17 January, 2020. The anonymized visiting records could be found at: http://www.watchman-pypi.com/ipOperation and the anonymized user information is available at http://www.watchman-pypi.com/ipAddress (i.e., we assume that an IP address corresponds to a unique visitor/user). We tried our best to filter out the visits by robots and crawlers, which have no functional operations on our website. Note that on `Watchman`'s homepage, we provided the tips to users for claiming that their interactions with `Watchman` may be recorded for research purposes. To protect the privacy of users, we will not disclose the data to third parties.

We observed that during the 40 days, `Watchman` had 1,648 visits involving 8,909 operations by 467 users, where the operations include diagnosing DC issues, displaying full dependency graphs (FDGs), and so on. Altogether, `Watchman` has generated diagnosis issue reports for 2,590 Python projects. Figure 1 shows the statistics of the daily visits during this time period. Obviously, the daily visits range from 19 to 59, and on average, there are 223 operations performed by 41 visits on each day. As shown in Table 1 and Figure 2, the above users are distributed in 15 different countries.

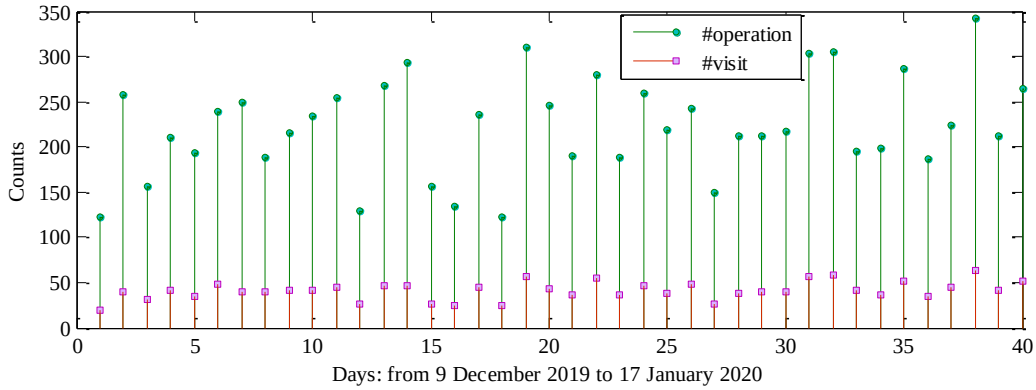

Figure 1. The distribution of the daily visits from 9 December, 2019 to 17 January, 2020

Table 1. The statistics of geographical locations of the visiting IP addresses

(the number of operations)

China	US	Australia	Canada	India	France	Japan	Singapore	UK	Russia	Brazil	Germany	Italu	South Africa	Rumania
7,625	572	88	173	138	63	11	74	71	21	8	5	25	30	2

Specially, during these days, 301 of 467 users (64.5%) are returning visitors to our website, for the purpose of diagnosing their projects. Figure 3 shows the number of visits of the above 301 returning users. We can see that 20 of them visited our website for more than 10 times on different days.

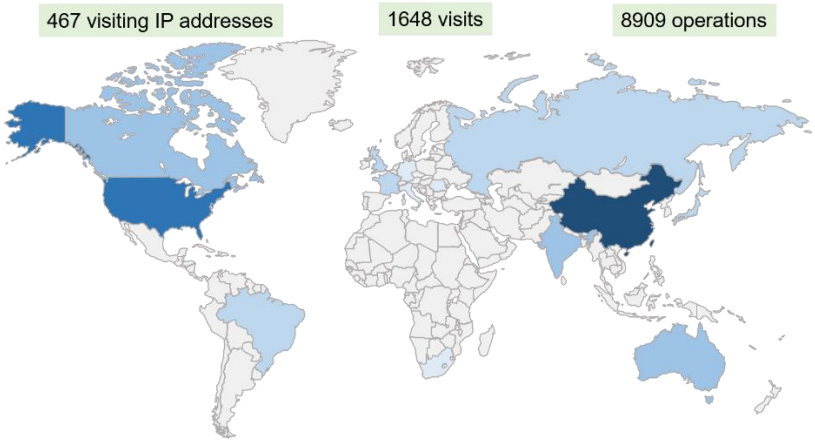

Figure 2. The geographical locations of the visiting IP addresses

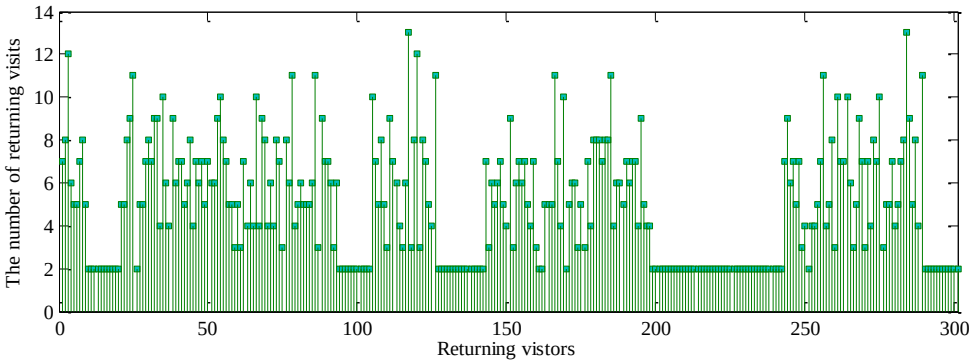

Figure 3. The number of visits of the 301 returning users

Table 2. The statistics of the user operations on `Watchman` website

#Diagnosing DC issues	#Displaying FDGs	#Saving FDGs in test file	#Demo	#About	#Go
1,717	2,721	1,016	1,286	734	1,435

Table 2 lists the statistics of the user operations on `Watchman` website. We can observe that most of the users are interested in displaying FDGs of their projects. The function of diagnosing DC issues attracts 1,717 operations, and many of them would like to save the FDGs for further analysis. To our surprise, 1,435 operations are for observing the topological structure of `PyPI` ecosystem and 1,286 operations are for watching our demo videos.

The above statistics demonstrate that real-world developers show interest in our technique. The returning visitors' behaviors indicate that the functionality of diagnosing DC issues provided by `Watchman` is useful and can be reproduced across different Python projects by different developers.

2. Reproduction reports from two software companies

`Neusoft` *Co. Ltd (SSE:600718)* is the largest IT solutions & services provider in China, and `Pinduoduo` *(Nasdaq:PDD)* is the fastest growing e-commerce startup in China. The two software companies have large-scale Python projects with dozens of third-party libraries. Diagnosing DC issues is one of the key challenges for their developers. We invited several experienced developers from these two companies to apply `Watchman` to their company projects. The invited developers from both companies confirmed the effectiveness and usefulness of `Watchman` on diagnosing DC issues. They also provided detailed reproduction reports and gave some feedback to us. Please refer to the appendix.

Appendix

1. A reproduction report from a developer of `Neusoft` *Co. Ltd (SSE: 600718)*

2. A reproduction report from a developer of `Pinduoduo` *Co. Ltd (Nasdaq: PDD)*

主　题:	Re: A request for your feedback on Watchman	
发件人:	"yangxue@neusoft.com" <yangxue@neusoft.com>	2020-1-23 14:07:40
收件人:	"王莹" <wangying@swc.neu.edu.cn>	

Dear Ying,

I am happy to provide my trial results of Watchman. Overall speaking, Watchman is a useful tool for detecting and predicting dependency hell problems. In most cases, the suggested repairing solutions are reasonable. The feedback on the questions are described as follows:

1. Which types of dependency conflict issues do you concern about most, in your development process?
   We encountered dependency conflicts many times, especially for the problems caused by the frequently updated libraries (e.g., urllib3 and numpy) and the downstream projects that always lock their dependent library versions (e.g., dash).

2. What is the most useful information generated by Watchman that can facilitate your diagnosing process for dependency conflicts? Why?
   I can prioritize the different types of information generated by Watchman, based on their usefulness for our development.
   1. **Diagnosis information**; 2. Repairing solutions; 3. **The online reachable dependency relationships** between our projects and the other ones. For example, Watchman can display that dependency trees of our downstream projects, which are useful for providing a comprehensive analysis method. 4. **The global network of PyPI ecosystem** on "Go" page, is also an interesting and useful function for me.

We used Watchman to analyze more than 50 Python projects, and have detected dependency bugs in 8 projects. I listed the following obtained results anonymously.

Table The trial results for your anonymous Python projects with DC issues

Projects	#Library (the number of referenced libraries)	Issue type	Whether the generated diagnosis information made sense?	Whether the generated fixing suggestions were being adopted?
P1	34	Build failures	Yes	Yes
P2	15	The project is affected by a library version being locked by downstream projects	Yes	Yes
P3	62	The project is affected by a library version	Yes	Yes

		that is close to its specified upper bound.		
P4	23	The project is affected by a library version that is close to its specified upper bound.	Yes	Yes
P5	6	Build failures	Yes	The generated solutions are reasonable, but I did not adopt them, as this library updated frequently and always cause build failures. Thus, I changed another similar library.
P6	93	The project is affected by a library version being locked by downstream projects	Yes	Yes
P7	27	Build failures	Yes	Yes
P8	44	The project is affected by a library version being locked by downstream projects	Yes	Yes

Thank you very much for your efforts on Watchman. Hope my report can help you. Please do not hesitate to contact me if you have any questions.

Best,
Xue

杨雪
资深软件工程师
医学影像产品部

医疗解决方案事业本部 临床医疗事业部
东软集团股份有限公司
沈阳市浑南区新秀街2号东软软件园
邮编：110179
手机：13998170276
邮箱：yangxue@neusoft.com
www.neusoft.com

Neusoft

From: 王莹
Date: 2020-01-23 09:58
To: yangxue@neusoft.com
Subject: A request for your feedback on Watchman

Dear friend,

We sincerely invite you to give us some feedback about the Watchman (**http://www.watchman-pypi.com/**) trials on your projects in terms of diagnosing dependency conflict issues. Before providing the trial results, could you please answer the following questions? With the help of your answers, we can evaluate and improve our Watchman. Thank you very much for your support and cooperation!

1. Which types of dependency conflict issues do you concern about most, in your development process?
2. What is the most useful information generated by Watchman that can facilitate your diagnosing process for dependency conflicts? Why?

Table The trial results for your anonymous Python projects with DC issues

Projects	*#Library*	*Issue Type*	*Whether the generated diagnosis information made sense?*	*Whether the generated fixing suggestions were being adopted?*

Looking forward to your response.
Best regards,
Ying

主　题:	Re:A request for your feedback on Watchman :-)	
发件人:	chiyuan <chiyuan@pinduoduo.com>	2020-1-24 0:21:58
收件人:	"王莹" <wangying@swc.neu.edu.cn>	

Dear Dr. Wang,

Thank you for your invitation of trials on Watchman. I would like to send you my feedback. My answers for your questions are listed as follows:

1. Which types of dependency conflict issues do you concern about most, in your development process?

   I concerned about the conflicts caused by the continual version updates on PyPI. Because in this ecosystem, all the developers will encounter a lot of projects with different goals in mind. For example, one can encounter a library that will be imported by other projects. I didn't have a solution to these conflicts other than checking conflict manually or using tools like pipdeptree or use other tools to install package like pipx, which seems to better handle these conflicts than pip.

   But watchman performs better than the state-of-the-art tools, as it can capture the library updates in the ecosystem on daily basis and analyze the all the affected downstream projects in the ecosystem to generate diagnosis information. In the PyPI ecosystem, once we get the message out about how to require these packages, this seems to me pretty easy to manage our project.

2. What is the most useful information generated by Watchman that can facilitate your diagnosing process for dependency conflicts? Why?

   The diagnosis information generated by Watchman and the function that shows the full dependency tree of a Python project. These provided information indeed helped us locate the root causes of dependency problems.

For the 43 analyzed Python projects, we found 11 projects have dependency problems. I removed the projects' names and listed the detection results in the following table.

Table The trial results for your anonymous Python projects with DC issues

Projects	#Library	Issue type	Whether the generated diagnosis information made sense?	Whether the generated fixing suggestions were being adopted?
A	56	Build failure introduced by a library update.	√	√
B	22	Build failure introduced by a library update.	√	√

C	75	Potential problem affected by a locked library version.	√	√
D	34	Build failure introduced by a library update.	√	The suggestion is acceptable. But, we migrated the conflicting library is into another one.
E	18	Potential problem affected by a locked library version.	√	√
F	93	Potential problem affected by a locked library version.	√	√
G	101	Potential problem affected by a locked library version.	√	√
H	56	Potential problem affected by a locked library version.	√	√
I	43	Build failure introduced by a library update.	√	The suggestion is acceptable. But, we migrated the conflicting library is into another one.
J	23	Build failure introduced by a library update.	√	The suggestion is acceptable. But, we migrated the conflicting library is into another one.
K	72	Potential problem affected by a locked library version.	√	√

If you need any other help, please let me know.

Best,

Jie Liang

---------- Original ----------

From: "王莹"<wangying@swc.neu.edu.cn>;
Date: Wed, Jan 22, 2020 10:12 PM
To: "chiyuan"<chiyuan@pinduoduo.com>;
Subject: A request for your feedback on Watchman :-)

Dear friend,

We sincerely invite you to give us some feedback about the Watchman (**http://www.watchman-pypi.com/**) trials on your projects in terms of diagnosing dependency conflict issues. Before providing the trial results, could you please answer the following questions? With the help of your answers, we can evaluate and improve our Watchman. Thank you very much for your support and cooperation!

1. Which types of dependency conflict issues do you concern about most, in your development process?
2. What is the most useful information generated by Watchman that can facilitate your diagnosing process for dependency conflicts? Why?

Table The trial results for your anonymous Python projects with DC issues

Projects	*#Library*	*Issue Type*	*Whether the generated diagnosis information made sense?*	*Whether the generated fixing suggestions were being adopted?*

Looking forward to your response.
Best regards,
Ying