Considerations for a future Open Code policy for NASA Space Science

Dana Akhmetova[1] and Jan Deca[2]

[1]KTH Royal Institute of Technology, Sweden, danieka@gmail.com
[2]Laboratory for Atmospheric and Space Physics, University of Colorado Boulder, USA, jandeca@gmail.com

1 The open source definition

The open source definition is set by the Open Source Initiative (OSI) [13] to determine the extent to which the software license meets the Open Software standards. In general, this definition implies that anyone can access, use, modify and distribute the open source in a changed or unchanged form without any restrictions, but under a license that provides a fixed set of freedoms defined by the OSI [12]. The code is defined as open source only if it is distributed under a license that is approved by the OSI. The license is approved by the OSI only if it meets the following ten points:

1. free redistribution (the license must not restrict selling/giving away the software as a part of an aggregated software package consisting of codes from other sources, neither request royalty/fee for such sale);

2. source code (the license must provide access to the software's non obfuscated source code in the preferred form in which a programmer would modify the program);

3. derived works (the license must provide the ability to modify the source code and distribute its derived products under the same terms as the license of the original software);

4. integrity of the author's source code (the license must ensure that the source code of the derived work is easily accessible, but may require its distribution in the form of the original source code together with patch files for the purpose of modifying the program at build time; it may require modified versions to have a name or version number easily distinguishable from the original code);

5. no discrimination against persons or groups (the license must not have any conditions that do not allow anyone to work on the code; in some countries there are export restrictions on certain types of software, and open source licenses may contain warnings of applicable restrictions and reminders of the duty to enforce the law; these licenses themselves, however, must not contain such restrictions);

6. no discrimination against fields of endeavor (the license must not restrict anyone from using the software in a specific field, including business or genetic research);

7. distribution of a license (the rights attached to the software must apply to all parties whom this software is redistributed to, without the need to comply with an additional license);

8. the license must not be specific to a product (the rights of the software must not depend on whether or not the software is a part of a particular software distribution);

9. the license must not restrict other software (e.g., other programs from the same distributive as the licensed code must not be forced to be open source);

10. the license must be technology-neutral (the distribution/availability of the license must not be based on any individual technology or style of interface).

Some aspects of the open source definition are often misinterpreted, leading to the following myths:

1. that open source codes are free to be used, changed, redistributed, at no charge;

2. that open source codes cannot generate profit;

3. that open source licensing implies that changes to the open source code must be shared publicly;

4. that the possibility to view the open source code is the same as the possibility to modify it;

5. that using open source components is equivalent to creating an entire system that is itself open sourced.

Today, the amount of projects with a publicly available code is booming, in part thanks to the contribution of software development platforms such as GitHub [10] that have provided a new norm for implementing software with an open code. Strictly speaking, the majority of GitHub projects do not fit the official open source definition. For example, in 2015 more than 80% of all GitHub projects did not have a license attached to their repositories [5]. Thus, stating that GitHub is a platform where developers open the source code of their projects, means exactly what GitHub explains in the "Licensing a repository" part of its

user documentation: "Public repositories on GitHub are often used to share open source software. For your repository to truly be open source, you'll need to license it so that others are free to use, change, and distribute the software... You're under no obligation to choose a license. However, without a license, the default copyright laws apply, meaning that you retain all rights to your source code and no one may reproduce, distribute, or create derivative works from your work" [11]. Hence, it is, in general, illegal to use/change/(re-)distribute code without a license even if it is publicly available through GitHub.

2 The iPIC3D plasma physics code

iPIC3D is a C++-based parallel (MPI plus OpenMP worksharing) particle-in-cell code, designed to simulate in detail multiscale space and fusion plasma environments [15]. To date it has been applied to various magnetic reconnection setups and the solar wind interaction with a variety of airless magnetized and non-magnetized bodies in the solar system. Next to pure scientific applications, including currently funded NASA research projects in plasma physics and planetary sciences, the code has been and is currently used as a pilot in a variety of European Exascale projects such as INTERTWinE [2] and AllScale [1]. The iPIC3D source code and its derivatives are available for download through the GitHub and BitBucket services [3, 4].

The iPIC3D code is licensed under the Apache License, Version 2.0 [8]. It is a permissive license (i.e., with minimal requirements for the code distribution) that:

- allows distribution, modification (the re-distributor does not have to release/open the modified source code), commercial use (the code and its modifications may be used for commercial motives), patent use (contributors give an express grant of patent rights to users) and private use of the code;

- requires preservation of license and copyright notices (they must be included with the code) and stating changes in the code (code modifications must be documented);

- thus, allows that licensed works, modifications, and larger works may be distributed under different terms and without source code.

In 2015, the Apache License was the third most popular license used by projects on GitHub [5]. The Apache License is approved by OSI [14], and therefore the iPIC3D code is open source.

As developers and users of iPIC3D, in what follows we elucidate on the possible advantages and disadvantages of an open code policy.

3 Considerations for an open code policy

We see the following advantages of an open code policy:

- the costs of implementing similar software are avoided, which will save (tax payers') money and decrease dependency on large IT vendors;

- the open code might discipline the developer, as all her/his work is visible for the public without restrictions, which may be an important factor in and motivation for self-development;

- NASA's open source policies ensure that its state-of-the-art developments are also accessible to other organizations;

- more effective joint code development with international collaborators can be accommodated: the use of open software, instead of a proprietary version, allows in many cases to overcome the difficulties associated with export control and provides greater flexibility in the use and modification of the code;

- using open source software, IT vendors are able develop their solutions quicker;

- sometimes end users do not need to have the source code itself, but just would like to have an opportunity to look at the code: this can be due to security reasons or, for example, due to their beliefs in the Free Software Movement [9] (which shares some points with the open source ideology);

- having an open code license may increase the success of a project (e.g., when the .NET Compiler Platform was made open source by Microsoft in 2014 [18]).

Thus, open source policies might provide more transparency, collaboration, flexibility and innovation.

Besides the open source approach, there are other options to license software: the code might not match the open source definition as defined by the OSI, but might still be publicly accessible (e.g., it is not allowed to modify the code, but it is allowed to check the code for its functionality and integrity). For example, the license of UnRAR [7], a utility for RAR archives, states: "UnRAR source code may be used in any software to handle RAR archives without limitations free of charge, but cannot be used to develop RAR (WinRAR) compatible archiver and to re-create RAR compression algorithm, which is proprietary. Distribution of modified UnRAR source code in separate form or as a part of other software is permitted, provided that full text of this paragraph, starting from "UnRAR source code" words, is included in license, or in documentation if license is not available, and in source code comments of resulting package" [6]. This means that UnRAR is a freeware - proprietary software that may be used free of charge, but its modification, redistribution, reproduction/reverse-

engineering without the author's permission is illegal [17].

In some situations the added value of an open source licence may not weigh up against its disadvantages. Therefore, in certain cases it may be better to adopt a proprietary license:

- simply making a code publicly accessible does not benefit anyone; beforehand the code should be re-factored such that it is reusable, in a good shape, structured, not overrun with bugs or specific hacks to the project, appropriately commented, and preferably have documentation and unit tests. In addition, the code developer should provide code maintenance. For example, she/he should be able to answer users' questions timely, react to bug reports, process git pull requests, build software packages for different platforms, support older versions of compilers, etc. All these actions require additional efforts from code developers and bring more responsibility at the same time. It is important to keep in mind that the core task of a (NASA) researcher is to deliver new science (of course always in a transparent and reproducible way);

- a successful business model for open source software is not always guaranteed. For example, the web server project NGINX eventually resulted in having a software subscription as the only viable development strategy [16];

- there are cultural barriers that make certain communities wary and hesitant about the need to make their codes open source;

- making a code open source might not always be worth the effort, as generally end users care more about the results the code produces, rather than whether or not the code is open source;

- providing technical support for a code that is not open source may be easier as the end user is not able to make any modifications to the code that may be tough to track and fix;

- the open source term no longer reflects how programmers develop software. It has become so widely used that it has lost its original meaning. Today open source merely points to a publicly available code with a license corresponding to the OSI. The original idea of open source (1990s) meant that a code does not have an owner and is managed by a license, which, in theory, would lead to more viable software. However, what should work in theory usually does not work in practice. It is almost common practice that a code is distributed free of charge under an open source license and subsequently redistributed/sold without reference to the original source. A proprietary code may provide more protection from unethical standards;

- although the scientific community as a whole might benefit from open source code, the scientist herself/himself might not: making her/his code freely available might give away the advantage from publishing new results with a large impact on her/his individual career to teams with more manpower and/or resources.

4 Conclusion

In conclusion, whereas programmers and software developers generally consider an open source policy as an opportunity, scientists may be more hesitant. Sharing their research tools might shrink the typically already narrow edge research teams have over their competitors. We would like to argue for an open source policy that is most efficient to produce transparent and reproducible science, but at the same time protects researchers within an already very competitive (scientific) environment.

References

[1] The AllScale european exascale project's official webpage. http://www.allscale.eu/home, 2017. [Online].

[2] The INTERTWinE european exascale project's official webpage. https://www.intertwine-project.eu/, 2017. [Online].

[3] The iPIC3D source code on BitBucket. https://bitbucket.org/bopkth/ipic3d-klm/src, 2017. [Online].

[4] One of the iPIC3D derivative's source code on GitHub. https://github.com/EPCCed/MPI_OpenMP_threads/tree/master/ipic3d, 2017. [Online].

[5] Ben Balter. The open source license usage on GitHub. https://github.com/blog/1964-open-source-license-usage-on-github-com, 2017. [Online].

[6] Peter Chapman. The license for use and distribution of the UnRAR source code. https://github.com/pmachapman/unrar/blob/master/license.txt, 2017. [Online].

[7] Peter Chapman. The UnRAR source code. https://github.com/pmachapman/unrar, 2017. [Online].

[8] The Apache Software Foundation. The Apache License, Version 2.0 of January 2004. https://www.apache.org/licenses/LICENSE-2.0, 2017. [Online].

[9] The Free Software Foundation. The Free Software Movement. https://www.gnu.org/philosophy/free-software-intro.en.html, 2017. [Online].

[10] GitHub. The GitHub software development platform. https://github.com/, 2017. [Online].

[11] GitHub. Licensing a GitHub repository. https://help.github.com/articles/licensing-a-repository/, 2017. [Online].

[12] The Open Source Initiative. The Open Source definition. https://opensource.org/osd, 2017. [Online].

[13] The Open Source Initiative. The Open Source Initiative home webpage. https://opensource.org/, 2017. [Online].

[14] The Open Source Initiative. OSI-approved licenses by name. https://opensource.org/licenses/alphabetical, 2017. [Online].

[15] Stefano Markidis, Giovanni Lapenta, et al. Multi-scale simulations of plasma with ipic3d. *Mathematics and Computers in Simulation*, 80(7):1509–1519, 2010.

[16] NGINX. The differences between open source NGINX and NGINX Plus. https://www.nginx.com/products/feature-matrix/, 2017. [Online].

[17] The Linux Information Project. The Freeware definition. http://www.linfo.org/freeware.html, 2017. [Online].

[18] Arfon Smith. The shape of open source on GitHub. https://github.com/blog/2195-the-shape-of-open-source, 2017. [Online].