BLAS and LAPACK runtime switching

Mo Zhou
lumin@debian.org

April 9, 2019

Abstract

Classical numerical linear algebra libraries, BLAS and LAPACK play important roles in the scientific computing field. Various demands on these libraries pose non-trivial challenge on system management and linux distribution development. By leveraging Debian's update-alternatives mechanism which enables user to switch BLAS and LAPACK libraries smoothly and painlessly, the problems could be properly and decently addressed. This project aims at introducing the mechanism into Gentoo's eselect framework to manage BLAS and LAPACK, providing equivalent or better functionality of Debian's update-alternatives.

1 Rationale

BLAS (Basic Linear Algebra Subroutines)[1] and LAPACK (Linear Algebra PACKage)[2] are important mathematical APIs/ABIs/libraries to performance-critical programs that manipulate dense and contiguous numerical arrays. BLAS provides low-level and frequently used linear algebra routines for vector and matrix operations; LAPACK provides higher-level functionality based on complex call graph over BLAS, hence BLAS is the most performance-critical part. These libnear algebra libraries are widely used in the scientific computing and many other areas, for example: Numpy, Scipy, Julia, Octave, R, Sagemath, Matlab, Mathematica and LINPACK [3] (High performance linpack is used to benchmark and rank supercomputers for the TOP500 list).

In practice, different users need different BLAS/LAPACK implementations with different configurations (e.g. threading model or index type variants) under different scenarios. Such extremely diversed and varied BLAS/LAPACK library requirement poses a non-trivial challenge to linux distribution developers.

Taking advantage from the update-alternatives mechanism of dpkg, Debian is able to address the challenge in a decent way. Debian users could switch the underlying BLAS/LAPACK implementations without recompiling their applications. At the same time, many BLAS implementations with different configurations could co-exist on the system without confliction. As a result, users could easily profile and benchmark different implementations efficiently, and painlessly switch the underlying implementation for different programs under different scenarios.

1.1 Solutions of Other Distros

Archlinux only provides the Netlib and OpenBLAS, where they conflict to each other. This won't satisfy all user's demand, because the OpenMP version of OpenBLAS is not a good choice for pthread-based programs.

Fedora's solution is to provide every possible version, and make them co-installable by assigning different SONAMEs. It leads to confusion and chaos if many alternative libraries co-exists on the system, e.g. libopenblas, libopenblasp, libopenblaso.

The BSD (Port) Family forces packages to use a specific implemtation on a per-package basis, which clearly doesn't satisfy the divsersed user demand.

1.2 Gentoo's Solution

Currently Gentoo's solution is based on eselect/pkg-config.

1. Gentoo's historical "alternatives.eclass" and the present "alternatives-2.eclass" are designed to mimic Debian's update-alternatives mechanism.

2. "alternatives-2.eclass" is still being tested and experimented in the science-overlay. Is still cannot be merged into Gentoo main repo because it doesn't fully comply with the requirements. Specifically, the eselect mechanism should not infect the build-time behaviour. This project will possibly reuse the code of "alternatives.eclass" and "alternatives-2.eclass".

3. The Gentoo community is still debating[4] on how to select BLAS and LAPACK. Some people think the update-alternatives mechanism is more flexible than the eselect mechanism, and hence Gentoo should adopt Debian's runtime-switch mechanism. Others argue that recompiling is inevitable after switch.

1.3 Proposed Solution

The solution of "BLAS and LAPACK runtime switching" is to mimic Debian, based on the eselect mechanism. The Netlib BLAS and LAPACK packages are mandatory when one needs to build any application on top of BLAS/LAPACK. The netlib implementation functions as the fallback implementation, a linkder stub, the standard header provider, and most importantly the fallback alternative for libblas.so.3 and liblapack.so.3. All packages depending on BLAS and LAPACK should link against libblas.so.3 and liblapack.so.3 respectively. They in turn could be provided by various 3rd party BLAS and LAPACK implementations. The recommended header used to build reverse dependencies is the one from netlib. All other BLAS and LAPACK implementations should register themselves as alternatives to the netlib implementation. Each implementation provides candidates for the shared objects, and their development files are stored in their own subdirectory.

2 Objective

Enable the users to switch the BLAS and LAPACK backend without re-compiling the reverse dependendies.

3 Deliverables

1. Develop eslect-blas and eselect-lapack integrations, with reference to Debian's update-alternatives, and the previous alternatives{,-2}.eclass.

2. Package BLIS for Gentoo, and register it as another netlib blas alternative.

3. Update the packaging of a BLAS/LAPACK reverse dependency to validate the efficacy of the mechanism on Gentoo. Specifically, the package will be compiled against BLIS's libblas.so.3 and netlib's libblas.so.3 respectively, and always use the netlib headers.

4. Modify the packaging for OpenBLAS to register libopenblas.so as an alternative to netlib blas. Header are not exposed in public include directory.

5. Update packaging for BLAS and LAPACK reverse dependencies, including Numpy, Scipy and R and enforce linkage against libblas.so.3 (any BLAS implementation could provide this). Note, Julia must be compiled against OpenBLAS. So Julia will also be tested to see whether the mechanism will have broken special packages such as Julia.

6. Upload (NEW) gentoo packaging for at least Netlib, OpenBLAS and BLIS.

7. Officialize BLAS/LAPACK reverse dependency packages for at least the most important ones, e.g. Numpy, Scipy, R and Julia.

8. Update gentoo wiki or documentations, describing related changes.

4 Timeline

I will be able to work full time during the ofcial GSoC time frame May 27, 2019 -August 19. 2019.

1. May 6 - May 27 (3 weeks): Get familiar with Gentoo development. Get to know the Gentoo science team.

2. May 27 - June 10 (2 weeks): Update packaging for blas-reference, cblas-reference.

3. June 10 - June 17 (1 week): Package BLIS for Gentoo, and register it as netlib alternative.

4. June 17 - June 24 (1 week): Update the packaging for several important packages such as Numpy, R and Julia, and conduct sanity tests regarding the mechanism.

5. June 24 - July 1 (1 week): Modify the OpenBLAS packaging and register it in the mechanism. Sanity tests will be carried out as well.

6. July 1 - July 22 (3 weeks): Update packaging for other important BLAS and LAPACK reverse dependencies, e.g. Octave and Scipy, and conduct sanity test.

7. July 22 - July 29 (1 week): Write documentation regarding to the changes.

8. Aug 5 - Aug 19 (2 weeks): Work with the Gentoo science team to stardardize the new mechanism and merge selected packages of the science overlay into Gentoo main repository. Fix all the rest BLAS/LAPACK reverse dependencies.

5 Biography

I'm a Debian Developer[5, 6] and a Master student at Xidian University, studying Artificial Intelligence, specifically Computer Vision and Deep Learning. I maintain many packages for Debian, including Julia, BLIS and MKL, and I'm recently working on updating Debian's OpenBLAS packaging. These packages are highly related to the GSoC proposal. I have planned to evaluate Gentoo for scientific research because the source-based installation mechanism allows users to install locally optimized packages and bypass many restrictions existing in Debian.

References

[1] https://en.wikipedia.org/wiki/Basic_Linear_Algebra_Subprograms

[2] https://en.wikipedia.org/wiki/LAPACK

[3] https://en.wikipedia.org/wiki/LINPACK

[4] https://github.com/gentoo/sci/issues/805

[5] https://nm.debian.org/person/lumin

[6] https://qa.debian.org/developer.php?login=lumin